LONDON'S GLOBAL UNIVERSITY
Department of Computer Science

Explainability in Transformer-Based Pre-Trained Language Models

Henning Heyen

Supervisor: Prof. Philip Treleaven
Co-Supervisor: Dr Maria Perez Ortiz, Dr. Adriano Koshiyama, Amy Widdicombe, Noah Siegel

A dissertation submitted in partial fulfillment for the degree
Master of Science

University College London
11 September 2023

Declaration

I, Henning Heyen, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

11/09/23

Date *Signature*

Abstract

This thesis presents original research on the relationship between the explainability of transformer-based pre-trained language models and model size. Performance increases smoothly with model size, as famously demonstrated by OpenAI's series of Generative Pre-Trained Transformers (GPT). However, there is a lack of understanding of how explainability is affected by increasing model size.

Our Methodology employs fine-tuned DeBERTaV3 models of four different sizes. We use Local Interpretable Model Agnostic Explanations (LIME) to explain the models' predictions. LIME is a widely used posthoc explainability tool that can be applied across various domains including natural language processing. This project considers two downstream tasks: natural language inference and zero shot classification.

Our Experiments and Results are compromised of two parts:

1. Model Evaluation: First, we replicate recent findings that performance increases with model size. The models are evaluated on three different datasets (MNLI, e-SNLI, CoS-e).

2. Explainability Evaluation: We investigate the quality of LIME explanations with respect to model size through two distinct quantitative approaches: faithfulness and plausibility. While faithfulness measures the extent to which the explanations reflect the model's decision process, plausibility assesses how well these explanations align with human explanations. For plausibility, we use two datasets that contain human annotated highlights (e-SNLI, CoS-e). We find that the faithfulness of LIME explanations improves with increasing model size in the zero shot classification setting and strongly varies in the natural language inference setting depending on the labels and on the predictions' correctness. Across all datasets plausibility is uncorrelated with model size suggesting misalignment between the posthoc explainations and the models' internal decision process.

The thesis contributes to the NLP community in the following way:

1. **Quantitative Evaluation of Explainability:** While there is no agreement on how to evaluate explainability quantitatively, this thesis reviews and applies two approaches (faithfulness and plausibility) for highlight-based posthoc explainability to state-of-the-art transformer models.

2. **Interplay between Explainability and Model Size:** Investigating the relationship between posthoc explainability and model size in state-of-the-art transformer models is novel research. We provide empirical evidence that the faithfulness of LIME explanations improves with model size in the zero-shot classification setting while their plausibility stays constant. We further find that explainability is label and prediction-dependent in the natural language inference setting.

3. **Extensible Code Repository**: All code to run the experiments is open-sourced[1]. It is designed to easily extend the scope by other models, datasets and explainability techniques to gain further insights into transformer explainability. An outline of future approaches and limitations will be discussed in depth.

In addition to these contributions, we are preparing to submit a paper at the NeurIPS 2023 workshop for Socially Responsible Language Modelling Research[2] (SoLaR).

[1] https://github.com/henningheyen/TransformersExplainability
[2] https://solar-neurips.github.io/

Acknowledgements

I would like to thank my supervisor Prof. Philip Treleaven for the continuous support and mentorship throughout this project. I wish to acknowledge Dr. Maria Perez Ortiz for the valuable feedback and for connecting me with Amy Widdicombe and Noah Siegel who perfectly combined expertise in explainability and language models and provided me with insights into the field that strongly shaped this work. Finally, I am grateful to work with Dr. Adriano Koshiyama from HolisticAI who encouraged me to pursue this research.

Impact Statement

This project is of particular interest to any stakeholder using transformer-based language models in high-stakes applications such as healthcare or credit scoring or any application in general that requires explanations on why the model's decision has been made. Additionally, regulators and policymakers might find this work useful as this project gives insights into how language models could comply with "right to explanation" policies. Lastly, developers can use the source code of this project to explain language models and objectively evaluate the explanations' quality.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The aim of this chapter is to provide an overview of this project by touching upon its motivation, research objectives, experiments, and contributions. The chapter briefly introduces transformer-based pre-trained language models and the research field of explainable AI and motivates the necessity of understanding the relationship between model size and explainability.

1.1 Motivation

AI systems powered by deep learning techniques disrupted many applications in both science and industry. Improvements in computational power and a large amount of training data enabled remarkable results in various domains. Examples include ResNet [15] surpassing human performance on image classification and AlphaFold [16] predicting the shape of proteins for drug discovery which was previously considered one of the most pressing open problems in biology [17]. A central limitation of most deep learning models is their dependency on large data corpora specific to the desired downstream task [18]. More recently, foundation models led to a paradigm shift in that models are pre-trained on a vast amount of broad data and can then be fine-tuned on a downstream task using a technique called transfer learning. This approach drastically reduces the amount of labeled data needed and avoids training models from scratch [18]. Additionally, the transformer architecture, first introduced in 2017 [1], significantly improved challenges in deep learning models regarding long-range dependencies in sequential data by proposing

the attention mechanism.

Especially, the Natural Language Processing (NLP) domain has benefited from applying transformer-based foundation models which approach or even surpass human-level performance in various tasks [19]. Research shows that performance in language models depends strongly on scale and less on model shape [6], where scale refers to the number of parameters, the training dataset size, and the amount of compute for training. Shape refers to architectural hyperparameters such as network width and depth. OpenAI famously pushed this phenomenon to an extreme with their series of Generative Pre-Trained Transformers (GPT). With every new version of GPT, for instance, the number of parameters and the amount of training data has increased. While GPT-2 has 1.5 billion parameters, GPT-3 has 175 billion parameters [20]. While OpenAI has not published any official information, it is believed that their GPT-4 model has about 100 trillion parameters [21].

While performance tends to improve with model complexity, model transparency suffers [22]. Due to the highly non-linear architecture, deep learning-based models are considered *black box* models, meaning the internal mechanism producing the model's output is opaque and not directly interpretable or understandable by humans. A common observation in generative large language models is that they are prone to hallucination, which refers to confident model outputs or responses that are not aligned with reality and that can not be justified by the training data [23]. Understanding and mitigating AI hallucinations is an ongoing scientific effort. Another open problem in large language models is the alignment problem which refers to "the challenge of ensuring that AI systems pursue goals that match human values or interests" [24]. Research shows that more capable systems and more affected by misaligned properties which poses the threat of super-intelligent systems undermining human control [24].

The hallucination and alignment problem illustrates that as AI systems will be increas-

ingly deployed in high-risk domains such as healthcare or autonomous driving, it becomes indispensable to understand *why* a certain decision has been taken by an AI model. Tackling this question has become its own research field commonly referred to as *explainable AI* [25]. Explainable AI can contribute to building trust in the interactions between AI systems and humans by verifying a system's functionality [26], identifying unintended behavior like biases and racial discrimination [27], providing accountability for wrong predictions [28] and avoiding maintenance costs [29]. While decision-making tasks are further automated by AI systems, explainable AI will also play a crucial role in compliance with future AI regulations. Even today, the EU General Data Protection Regulation (GDPR) states the "right to explanation" [30], which also applies to complex deep learning systems. The consequences of deploying non-explainable AI in high-risk domains can be significant. For example, the machine learning algorithm COMPAS was actively deployed in the U.S. justice system for predicting the risk of recidivism and was found to be biased against black people [31]. Explainable AI may have prevented the deployment [32].

Black box models, such as neural networks, are considered intrinsically intransparent. However, posthoc explainability techniques have been developed to explain neural networks across various domains, including NLP [22]. While these techniques have been applied to transformer models [5], there is no literature on the interplay between explainability and transformer model size to the best of our knowledge. Additionally, there is no consensus on how to quantitatively measure posthoc explanations in NLP [22]. Some approaches have been found to even contradict each other [33]. As developers will likely continue to scale up model size to boost performance it will be crucial to understand and effectively measure explainability to ensure safe and trustworthy future developments in AI.

1.2 Research Objectives

This thesis aims to empirically investigate the relationship between model size and post hoc explainability. We define model size as the number of parameters in the neural network. First, we intend to replicate that model performance increases with model size using two downstream tasks. Then we conduct experiments on the explainability of these models and provide two simple approaches to evaluate the explanations (faithfulness and plausibility). As the scope of this project is limited to only one explainability technique (LIME) and to one transformer-based pre-trained model of different sizes (DeBERTaV3), the goal is to also provide an extensible repository for researchers to extend this work by additional explainability techniques, models and datasets.

1.3 Research Experiments

For empirically exploring the relationship between model size and explainability, the experimental setup includes four different pre-trained DeBERTaV3 models that are publicly available on Huggingface [34]. The backbone number of parameters ranges from 22 million to 304 million. Using fine-tuned versions of those four models, we perform two downstream tasks: Natural Language Inference (NLI) and Zero-Shot Classification on three datasets, i.e. MNLI [11], e-SNLI [10] and CoS-e [14]. Our experiments consist of two parts 4:

1. Model Evaluation: First, all four models are evaluated on performance. We report the common classification metrics accuracy, recall, precision, and f1 scores. We find that for all three datasets, performance increases with the model size.

2. Explainability Evaluation: We then apply LIME to all four models on the different datasets. We evaluate the explanations based on their faithfulness and plausibility. Faithfulness measures the extent to which the explanation reflects the models'

decision process. The metrics used to measure faithfulness are comprehensiveness (removing the explanation from the input sequence) and sufficiency (using the explanation itself for prediction). By the change in prediction, we can assess the models' faithfulness. Plausibility, on the other hand, expresses the agreement with human ground truth explanations (available in e-SNLI and CoS-e). We measure plausibility by calculating the intersection over union (IOU) and token level f1 (TokenF1) scores between the generated and human explanation.

(a) Local Evaluation: First, the LIME explanations are investigated on a local level to provide an intuition for the faithfulness and plausibility metrics. We use instances from CoS-e and e-SNLI.

(b) Global Evaluation: To draw conclusions about the models' explainability we scale the number of LIME explanations up to 100. We report on the metrics for each task across all four model sizes. We find that faithfulness increases with model size in the zero-shot classification setting while faithfulness depends on the labels and predictions in natural language inference. The plausibility metrics are uncorrelated to model size.

1.4 Scientific Contributions

There is no agreed framework for evaluating posthoc explanations quantitatively. This project thoroughly reviews explainability metrics and applies two different approaches (faithfulness and plausibility) for LIME explanations on transformer-based language models.

To the best of our knowledge, investigating explainability with respect to model size for pre-trained language models is novel research. With this project, we provide the first empirical evidence that there is a misalignment between posthoc explanations and the model's internal decision process as plausibility stays constant while model performance

increases with model size. We further find evidence that faithfulness is task-dependent.

This project is by no means complete, and more experiments including other explainability techniques and models are required to better understand the relationship between explainability and model size. Therefore, the thesis is accompanied by an extensible Python repository[1] which allows stakeholders to experiment with further models, datasets, and explainability techniques. We review other explainability techniques and models and provide an in-depth discussion for potential future work.

1.5 Thesis Structure

The structure of this thesis is organised as follows:

Chapter 2 - Background and Literature Review: First, relevant literature and the key concepts in this research are reviewed to introduce the reader to the environment of this thesis. The chapter begins with a review of transformers including attention and pre-training. Then explainability in the domain of NLP is surveyed, starting with a hierarchy of explainability and different types of NLP explanations followed by explainability metrics. The explainability technique used in this thesis (LIME) will be introduced in more detail. The chapter ends with a review of literature that is more specifically associated with the relationship between language models and explainability.

Chapter 3 - Methodology: This chapter provides an overview of the experimental setup. We report on the models, tasks, datasets, explainability technique, and metrics used in the experiments.

Chapter 4 - Experiments and Results: This chapter describes the conducted experiments. We first replicate findings regarding the performance boost for larger models and then

[1] https://github.com/henningheyen/TransformersExplainability

evaluate the LIME explanations on both a local and global level. The experiments are accompanied by result tables and figures.

Chapter 5 - Discussion: Then, the results are critically analysed. We further discuss the limitations of the experiments followed by an in-depth review of potential future work and inspirations to extend our experiments.

Chapter 6 - Conclusion: Finally, we summarize the experiments and the key findings. Lastly, the thesis is put in a broader context by discussing ethical and sustainability considerations.

Chapter 2

Background and Literature Review

This chapter serves as a survey on transformers and explainability. First, we review the literature regarding transformers, including the attention mechanism and pre-training. Then, AI explainability is introduced and broadly categorized. Explainability types and metrics in the domain of NLP are described in more detail. Finally, we review related literature regarding the interplay between transformer-based language models and explainability.

2.1 Transformer-Based Pre-Trained Language Models

Transformer-based models have achieved human-level performance across various domains, including computer vision, speech, audio, and natural language [35]. This section serves as a primer on transformers for NLP. We discuss the NLP pipeline, the original transformer architecture, the attention mechanism, and pre-training.

2.1.1 NLP Pipeline using Neural Networks

This section is based on "A Survey of the Usages of Deep Learning for Natural Language Processing" [36]. NLP involves building computer systems for understanding human language. Processing language statistically is considered inherently challenging because of the informal nature of language (e.g. ambiguity, synonyms). Classical NLP tasks include sentiment analysis, machine translation, question answering, or summarization. Improve-

ments in parallel computing and computational power have almost entirely replaced traditional machine learning (ML) approaches by neural networks. Neural networks in NLP are generally based on an encoder-decoder architecture. The encoder first tokenizes the raw input sequence into tokens (e.g. words) which are then embedded into a numerical representation. The decoder then classifies the numerical representation according to a downstream task. (e.g. classifying positive or negative sentiment). Both components can be neural networks whose parameters are trainable via backpropagation. Due to the sequential nature of text containing complex dependencies, recurrent neural networks (RNNs) were proposed to incorporate a memory mechanism.

2.1.2 Transformer Architecture

The central limitation of RNN-based models, such as Long Short Term Memory (LSTM) and Gated Recurrent Networks (GRU) [37], are their inherently sequential computation which precludes parallelization and scalability [38]. However, both are crucial to capturing higher-level semantics in long sequences of data with long-range dependencies. Transformers, first introduced in 2017 with the paper "Attention is all you need" [1], proposes the attention mechanism, which allows for global dependencies between input and output and parallelized computation. Transformers are neural networks designed for sequential data such as text or audio. The original study uses a sequence-to-sequence transformer for machine translation. As depicted in figure 2.1, the original transformer architecture consists of two blocks, the decoder and the encoder.

On a high level, the encoder encodes the sequence (e.g. text) into a numerical representation. Each token in the sequence (e.g. words) corresponds to one numerical vector representation. The dimension of these vectors is called hidden size or model dimension and is defined by the architecture of the network. The hidden size in the original transformer is 512. The embedding is a contextualised representation of the input sequence, meaning each token vector contains a context within the sequence, which is achieved by

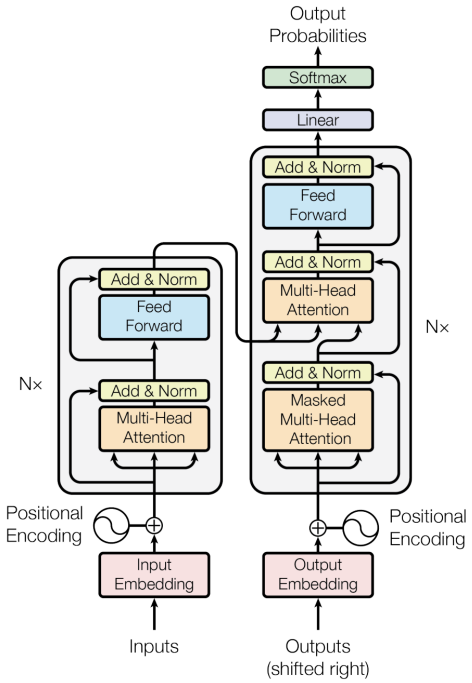


Figure 2.1: Original transformer architecture [1].

the self-attention mechanism. As each token is contextualised with respect to both the subsequent and previous token, the attention mechanism in the encoder is considered bi-directional. More precisely, the encoder consists of an input embedding followed by N identical layers, where each layer is composed of a multi-head attention layer and a feed-forward layer. To allow deeper networks, residual connections [15] are employed, followed by layer normalization [39].

Similarly, the decoder consists of N identical layers with the same sublayers. However, the decoder is prefixed by a masked multi-head attention layer. The context to the right of the current token is masked, which ensures that the self-attention layer is only attending to previous tokens. Therefore, the decoder is unidirectional. The decoder takes the contextualised representation from the encoder as input. Then the decoder generates an output sequence which also serves as the input of the decoder for predicting the next token. The process repeats until the end of the sequence (EOS) token is predicted. This

process is known as autoregression. A high-level description of the original transformer architecture is summarised in figure 2.2.

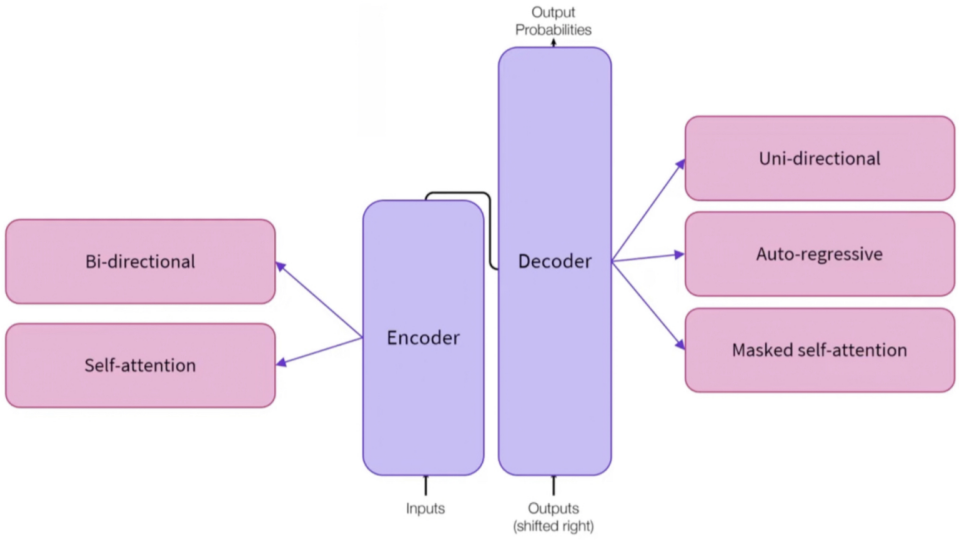


Figure 2.2: Transformer high-level architecture from Huggingface [2].

2.1.3 Attention Mechanism

The key concept of transformers is the attention mechanism. It enables representing dependencies and correlations within a sequence. For example, consider the sequence or words: *Tom loves maths. He is good at it.* Applying self-attention would result in a contextualised numerical representation for each word (e.g. *He* relates to *Tom*, and *it* relates to *maths*). While we cover the mathematical expressions for the attention components, detailed derivations and illustrations are out of scope for this thesis. A well-organized tutorial on further details can be found here [40].

"An attention function can be described as mapping a query and a set of key-value pairs to an output" [1]. Queries, keys and values are matrices containing a query vector, key vector and value vector for each token in the sequence. The three matrices are calculated by multiplying the input embedding by three trainable weight matrices. The scaled dot product attention layer can be expressed as follows [1]:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D_k}}\right)\mathbf{V} = \mathbf{A}\mathbf{V} \tag{2.1}$$

Where $\mathbf{Q} \in \mathbb{R}^{N\times D_k}$, $\mathbf{K} \in \mathbb{R}^{M\times D_k}$ and $\mathbf{V} \in \mathbb{R}^{M\times D_v}$ are the queries keys and values matrices. N denotes the length of the queries, and M denotes the length of the keys and values. D_v and D_k are the dimensions of values and keys (or queries), respectively. The dot product between $\mathbf{Q}$ and $\mathbf{K}$ is divided by $\sqrt{D_k}$ to obtain more stable gradients to mitigate vanishing gradients [38]. The softmax function projects the resulting matrix onto the unit scale with a sum equal to 1. Finally, the resulting attention matrix $\mathbf{A}$ is multiplied by the value matrix $\mathbf{V}$.

Instead of simply applying a single attention function, most transformers use multi-head attention [35], where the dimension of the original queries, keys and values are projected from the model dimension D_m onto D_k, D_k and D_v respectively with H different trainable projections. For each of the projections, the attention output is calculated according to equation (2.1). The outputs are concatenated and projected back to the model dimension D_m:

$$\begin{aligned}\text{MultiHeadAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \text{Concat}(head_1, \ldots, head_H)\mathbf{W}^O \\ \text{where head}_i &= \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)\end{aligned} \tag{2.2}$$

Where $\mathbf{W}_i^Q \in \mathbb{R}^{D_\text{m}\times D_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{D_\text{m}\times D_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{D_\text{m}\times D_v}$, and $\mathbf{W}^O \in \mathbb{R}^{(HD_v)\times D_\text{m}}$. The original transformer architecture uses $H = 8$ heads with $D_k = D_v = \frac{D_m}{H} = 64$. Even though the number of attention layers increases, computational costs stay similar due to the dimension reduction [1]. An illustration for both single and multi-head attention is depicted in figure 2.3.

The original architecture uses two different types of attention:

1. **Self-Attention** is deployed in the encoder by setting $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{X}$,where $\mathbf{X}$

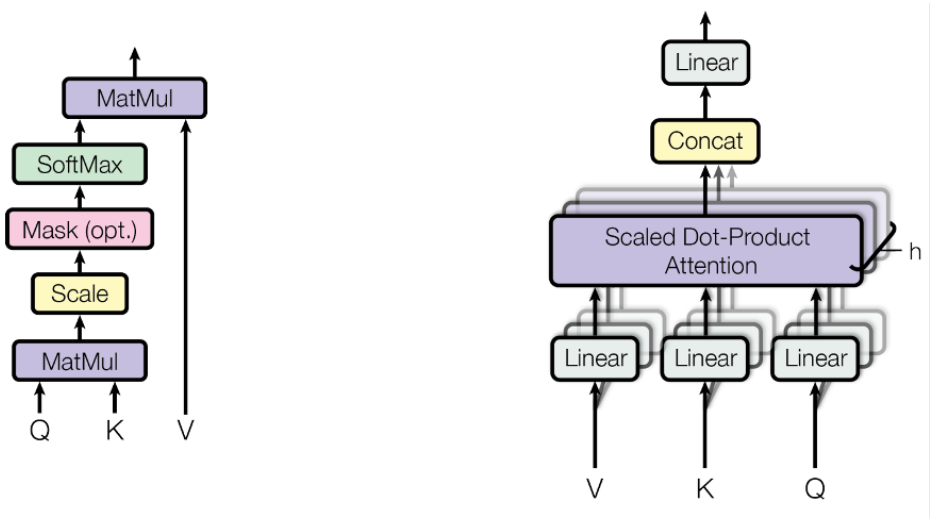


Figure 2.3: Scaled Dot-Product Attention (left). Multi-Head Attention (right) from [1].

is the output of the previous layer. Here the queries are attending to all key-value pairs (bidirectional attention).

2. **Masked Self Attention** restricts the decoder such that the queries at each position can only attend to key-value pairs up to the current position (unidirectional attention). This can be achieved by applying a mask function to the unnormalized attention matrix $\mathbf{A}' = \exp(\mathbf{Q}\mathbf{K}^T/\sqrt{D_k})$ where for all masked positions $A_{ij} = -\infty$ if $i < j$.

Note that, unlike RNNs, the attention mechanism itself is permutation invariant, meaning changing the token position would yield the same attention matrix. This *bag of word* approach would neglect the positional importance of the token. For example, *I Am Hungry* would result in the same attention matrix as *Am I Hungry* which would miss important semantic differences. Therefore, the so-called positional encodings take the relative and absolute position of each token in the sequence into account. The positional encoding is injected between the embedding and the encoder or decoder stack, respectively, such that the two can be summed up. The original work uses handcrafted positional encodings, but they can also be learned as in BERT [41].

2.1.4 Pre-trained Transformers

RNNs rely on the locality of underlying data (e.g. order of words in a sentence). Transformers, however, do not make any structural assumptions. While this makes transformers prone to overfitting if data is limited [35], they can learn universal representations when a large amount of data is available. Therefore, transformers have been widely used as pre-trained models. Pre-training refers to training a model on large corpora of raw data (e.g. text) in a self-supervised fashion to learn a universal statistical understanding of the data without specifying a practical downstream task. In NLP, the self-supervised objective is often to predict a masked word in the input text. This allows training on vast amounts of text data without any human-annotated labels [42]. Once pre-trained, these models can be fine-tuned for specific downstream tasks (e.g. sentiment analysis [43]). Fine-tuning typically involves adapting the pre-trained model's output layer to the new prediction task on a smaller, task-specific dataset (e.g., SST [44]). This practice of leveraging pre-trained representations from one task and applying it to another is termed transfer learning.

A central advantage of transfer learning is that, if applicable, a model does not need to be trained from scratch, which drastically saves time and computational resources. Additionally, transfer learning can be more data efficient and can increase performance in domains where labelled data is limited [18].

Pre-training has been established before the breakthroughs with transformers. The most common pre-trained models in NLP were Skip-Gram [45] and GloVe [46]. Even though they can capture some semantic meanings of language, they fail to capture higher-level linguistic concepts like ambiguity and syntactic structures [42]. Regarding performance benchmarks in NLP, transformers have undoubtedly led to significant improvements. All top ten models in the General Language Understanding Evaluation (GLUE) benchmark [47], for instance, are based on large pre-trained transformer models [19]. A possible

explanation why transformers perform generally better on large amounts of data is because transformers have fewer assumptions on the data structure compared to RNNs [35], which makes them more scalable in general. Transformer-based models can be grouped into the following broad categories:

Encoder-Only transformers only use the encoder block of a traditional architecture. The most widely used encoder-only representative is the Bidirectional Encoder Representations from Transformers (BERT) [41]. Encoder-Only models are mainly used for natural language understanding and sequence classification tasks. As we use a BERT-based model in the experiments, the model is covered in more detail.

The original BERT base model has 12 layers with self-attention (with 12 heads each) and a hidden size of 786, resulting in 110 billion trainable parameters [41]. The authors also introduced a large BERT model using 24 layers, 1024 hidden size and 16 heads resulting in 340 million parameters. BERT uses WordPiece tokenization [48] and was trained on two self-supervised objectives: Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). In MLM, the model had to predict a set of randomly masked tokens from the input text, and in NSP, the model classified whether sentence B follows sentence A. Half of the time, sentence B follows sentence A. For the other half, sentence B is a random sentence. BERT was trained on 800 million words of the BookCorpus [49] and 2.5 million words of English Wikipedia (16GB in total). BERT was evaluated on eleven different NLP benchmarks, including GLUE (79.6 base, 82.1 large) [41]. Several modifications of BERT have been introduced to lower memory consumption, see ALBERT [50], to improve general performance, see RoBERTa [51] and DeBERTa [52], and to reduce the number of parameters, see DistilBERT [53].

Decoder-Only models are mainly used for generative tasks like text generation[1], i.e. predicting the next token in a sequence. The most prominent representative decoder-only

[1]Text generation is also called language modelling, but we find it confusing as we use the term language models task independently. We aim to be consistent.

model is OpenAI's Generative Pre-Trained Transformer (GPT) series, including GPT-1 [54], GPT-2 [55] and GPT 3 [20]. Open AI's user interface chatGPT [56] already offers GPT-4. However, apart from a technical report [57] covering safety aspects of GPT-4 no official details regarding the architecture of GPT-4 have been published so far.

Encoder-Decoder, also called sequence-to-sequence models, correspond to the original transformer architecture and use both building blocks. The models are best suited for generative tasks that depend on an input sequence, such as translation, summarization, or question-answering. Encoder-Decoder models are mostly applied for problems that require both natural language understanding and generation. Among the most used sequence-to-sequence, pre-trained models are BART [58] and the Text-To-Text Transfer Transformer (T5) [59]. All models mentioned in this section are presented on a timeline in figure 2.4.

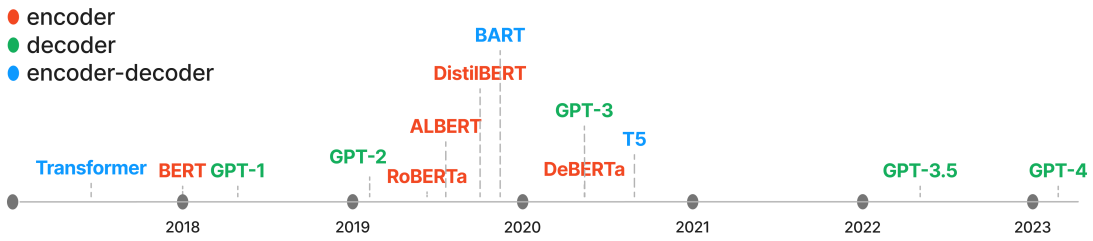


Figure 2.4: Chronology of the pre-trained transformers discussed in section 2.1.4.

2.2 Explainability in NLP

Motivated from a social science perspective, explainability can be defined as "the degree to which an observer can understand the cause of a decision" [60]. As AI is increasingly deployed in high stake applications such as autonomous driving or healthcare, understanding *why* a decision was made is crucial. Explainability can not only contribute to

AI safety [26], it can help to detect biases such as racial discrimination [27], avoid maintenance costs [29], provide accountability for wrong predictions [28] and comply to AI regulations such as GDPR's "right to explanation" [30].

This section reviews the literature on AI explainability by first sketching a general hierarchy of explainability techniques. Then different types of explanations in the domain of NLP are introduced. While we touch on a broad variety of techniques on a high level with references throughout we introduce LIME in more detail as LIME is used in the experiments. Lastly, explainability metrics are discussed with a focus on the metrics used in the experiments (faithfulness and plausibility).

2.2.1 Hierarchy of Explainability

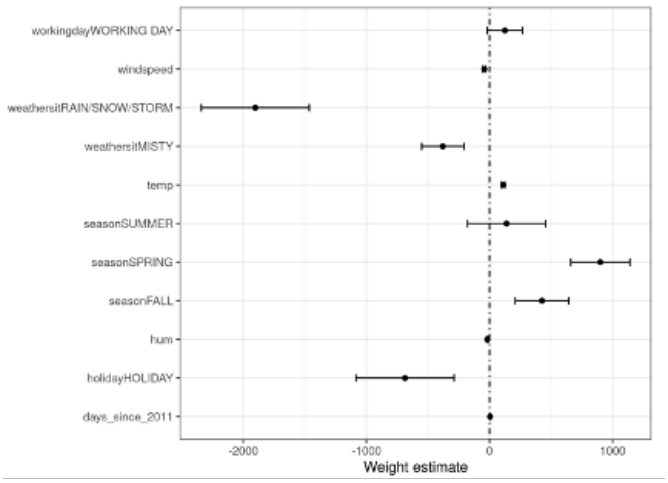
In the literature, the terms *explainability* and *interpretability* are frequently used interchangeably. However, some studies draw a distinction between the two (e.g. [61]). For the scope of this work, we have chosen to just use the term *explainability* consistently.

Although nuances exist, explainability techniques can be broadly categorised along two axes: Intrinsic vs. Posthoc and Local vs. Global [3]:

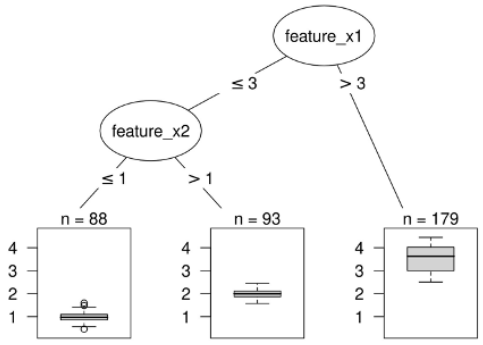
Intrinsic vs. Posthoc:

Intrinsic or white-box methods refer to techniques that rely on the model's inherently transparent architecture. The complexity of an intrinsically explainable model is often very limited but enables humans to understand the internal logic of a model. Intrinsic methods depend on a specific model and are, therefore, considered model-specific. For instance, in linear regression, the regression weight w_i of some feature x_i can be directly translated into the effect of that feature on the outcome variable y when all other features are fixed. Note that this conclusion only holds if the assumption of linear regression are met [3]. Another class of intrinsic models are decision trees. The explanation can be directly taken from the decision tree generated. For example, if feature $x_1 > 3$, the

model predicts the third class. Furthermore, feature importance scores can be calculated on decision trees as well [3]. An example of linear regression and a decision tree is illustrated in figure 2.5.



(a) Linear regression weights with 95% confidence intervals for predicting the number of bike rentals.



(b) Decision tree with synthetic data, e.g. explains that samples with $x_1 > 3$ are predicted the third bin.

Figure 2.5: Examples for intrinsically explainable models from [3].

Posthoc methods, on the other hand, are applied after the model has been trained. That means the explanations cannot be obtained from the model directly, but an extra step has to be taken. If the method does not make any assumption about the model's architecture, it is considered a model-agnostic method. Model-agnostic techniques only require the model's prediction and are, therefore, very flexible as they can be applied to any machine learning model.

Global vs. Local:

The other axis considers whether the technique explains a single prediction or the entire model. Intrinsic techniques normally allow both global and local explanations, as the weights can be extracted for single instances as well. Posthoc methods, however,

are often designed to either explain the model globally or explain single instances. Local posthoc techniques can be further grouped by perturbation-based (LIME, Anchors, SHAP), counterfactual-based (counterfactual explanations) and gradient-based (Vanilla Gradient, Integrated Gradient, Smoothgrad). We describe these local posthoc explainability on a high level in the following:

- **LIME** (Local Interpretable Model-Agnostic Explanations) [4] approximates a local prediction of a black box model by learning a more intrinsically explainable so-called surrogate model. As LIME will be applied in the experiments, the method will be introduced in more depth in section 2.2.3.

- **Anchors** [62] work similar to LIME. However, instead of approximating by a linear model, this method learns a local region where changes in the feature values do not affect the prediction. Anchors utilize reinforcement learning techniques and graph search algorithms.

- **SHAP** (Shapley Additive Explanations) [63] are based on the coalitional game-theoretic concept of Shapley values [64], where the feature values act as players, and the prediction acts as the payoff.

- **Counterfactual Explanations** [65] are based on observing how a prediction behaves if the feature values of a particular instance change. The goal of counterfactual explanations is to find a feature value where the prediction flips. For example, if the income of a person had exceeded 50,000$, then the model would have predicted a higher credit score.

- **Vanilla Gradient** [66] is the simplest form of gradient-based techniques which are widely applied in neural networks. They cannot be classified as model-agnostic as the model has to be differentiable. However, they apply to most neural network architecture and are therefore also not considered model-specific. Vanilla gradient computes the gradient of the output prediction with respect to the input features

for a specific instance. The gradient can suggest which input features affect the output the most. This often results in a sparse, non-robust score vector.

- **Integrated Gradient** [67] improves on vanilla gradient by starting from a baseline input (e.g. PAD tokens) and summing over all gradients that lead to the original input, which yields more robust vectors.
- **Smoothgrad** [68] has been successfully applied to image classifiers [68] but can also be used in NLP [69]. The idea is to repeatedly add random noise to the input sequence and average over the calculated gradients, which leads to more stable and less noisy score vectors.

Common global posthoc explainability techniques include:

- **PDP** (Partial Dependency Plots) [70] are a visualisation techniques to show the dependence between features and target variable. As human visual perceptions to limited to three dimensions, this would normally only include one or two features. PDPs assume that the features are independent, which is often violated in real-world applications.
- **PFI** (Permutation Feature Importance) [71] is a technique to observe how the model error changes if the feature values are perturbated. For example, if shuffling the values of a feature leaves the error unchanged, it is considered not important.

The above-mentioned techniques can be sketched with respect to the two axes as in figure 2.6.

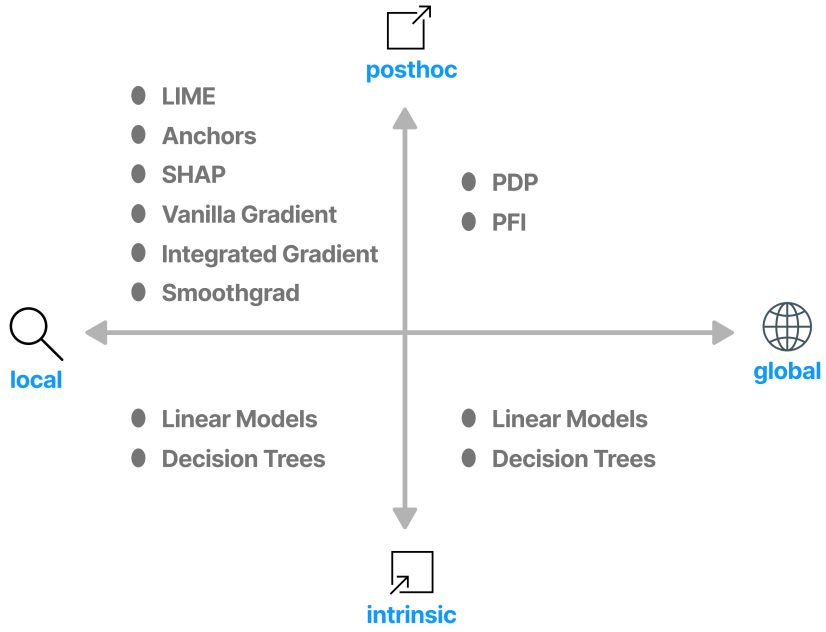


Figure 2.6: Hierarchy of explainability techniques as discussed in section 2.2.1.

2.2.2 Types of Explanations in Natural Language Processing

The type of explanation refers to how the explanation is conveyed to humans. For tabular data, the explicit feature contributions can serve as an explanation (e.g. salary in credit scoring). For unstructured text data, the type of explanation is not as straightforward. In this section, we identify two different types of explanations for NLP: highlight-based and natural language explanations [72].

Highlight-Based:

This type of explanation, also referred to as saliency-based, highlights the input features (tokens in NLP) that are most important to the model's prediction. Highlight-based techniques are always local as tokens from a specific input sequence are highlighted. Highlights can only be computed with respect to a predicted class. We can mathemati-

cally express a highlight-based explanation as follows [22]:

$$E(\mathbf{x}, c) \colon I^d \to \mathbb{R}^d \tag{2.3}$$

where $\mathbf{x}$ is the input sequence, c is the predicted class, I is the input domain (strings for text) and d is the input dimension (number of tokens). Since the outputs are scalars, the explanations are quantitative in nature. Highlight-based techniques are therefore considered rather functionally grounded than human grounded.

Highlight-based techniques include LIME, Anchors, SHAP and Gradient-Based. All of these methods follow the mapping from the input feature space to a real-valued score vector of the same dimension. A LIME example is depicted in figure 2.7

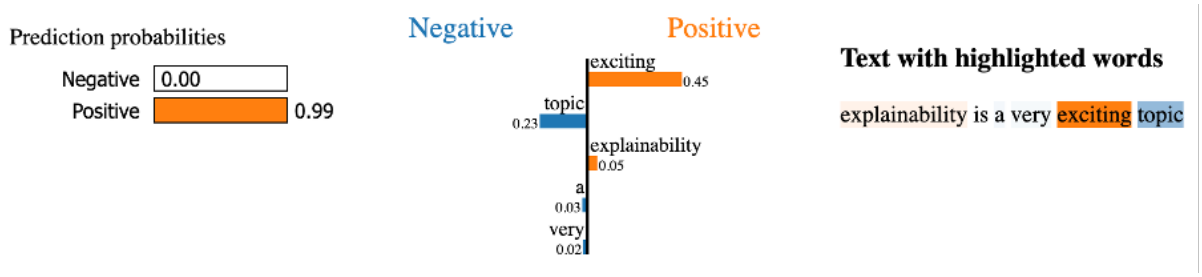


Figure 2.7: Sentiment analysis LIME example.

Note that attention weights in transformer models have also been studied as a highlight-based technique to explain the model's prediction [73]. However, several studies argue that attention does not provide meaningful explanations. The study "Attention is not Explanation" [74], for instance, identified different attention distributions yielding equivalent predictions. Another study states using attention for explainability is "by no means a fail-safe indicator"[75]. Therefore, attention is not considered an explainability technique in this thesis.

Natural Language based:

While highlight-based techniques are functionally grounded, they are considered less human-understandable. Therefore, it is tempting to use natural language or free-text explanations. They are generally more expressive and readable than highlights. Attempts have been made to leverage natural language explanations to improve the performance of NLP systems. The work of [10], for instance, extended the Stanford Natural Language Inference (SNLI) dataset by natural language explanations (e-SNLI) to train models for generating natural language explanations. More and more efforts are made to label existing datasets with natural language explanations [72]. However, natural language explanations are costly, and their correctness is hard to verify. There is no framework to evaluate natural language explanation generated which is why they are often qualitatively evaluated by humans [10].

This study focuses on highlight-based explanations due to their functional groundless. More precisely, we use LIME in the experiments. The following section will introduce LIME in more detail.

2.2.3 Local Interpretable Model Agnostic Explanations (LIME)

LIME, introduced by [4], is a local model agnostic explainability technique that can be applied to a wide variety of data modalities (text, images, tabular). The intuition behind LIME is to approximate the model prediction by an intrinsically explainable model and use its weights as feature importance scores.

Figure 2.8 illustrates the high-level idea behind LIME for a binary classification task. The complex decision boundary of the black box model is depicted by the red and blue background. Assume the bold red cross is the instance we want to explain. LIME generates perturbed instances near the instance, predicts the corresponding labels using the black box model, and weighs the predictions by the proximity to the instance as

indicated by size. The generated sample-label pairs are then used to train a linear model that is locally faithful to the original prediction. The weights of the trained linear model correspond to feature importance scores.

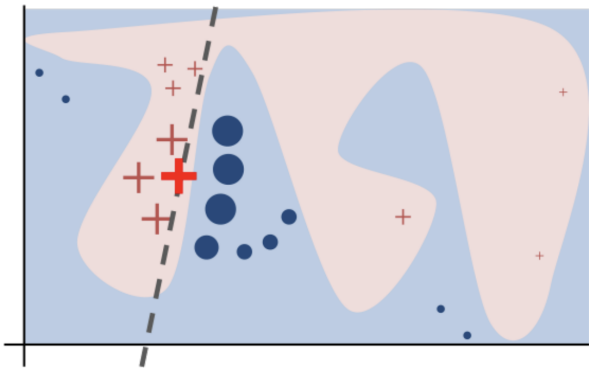


Figure 2.8: Intuition behind LIME borrowed from [4].

The original paper [4] uses a logistic regression model with L1 regularization (LASSO). LIME can be formalized according to equation as follows [22]:

$$E_{\text{LIME}}(\mathbf{x}, c) = \underset{w}{\text{argmin}} \left(\frac{1}{k} \sum_{i=1}^{k} (p(c|\tilde{\mathbf{x}}_i; \theta) \log(q(\tilde{\mathbf{x}}_i)) + (1 - p(c|\tilde{\mathbf{x}}_i; \theta)) \log(1 - q(\tilde{\mathbf{x}}_i))) + \lambda \|w\|_1 \right)$$

$$\text{where } q(\tilde{\mathbf{x}}) = \sigma(\mathbf{w}\tilde{\mathbf{x}}) = \left(\frac{1}{1 + e^{-\mathbf{w}\tilde{\mathbf{x}}}}\right) \tag{2.4}$$

In the equation, $\mathbf{x}$ is the instance we want to explain with respect to class c. The $\tilde{\mathbf{x}}_i$ are the perturbated samples. The original work uses samples from the bag-of-word representation with cosine distance. $p(c|\tilde{\mathbf{x}}_i; \theta)$ is the black box prediction for class c on the perturbated sample $\tilde{\mathbf{x}}_i$ given the black box model weights θ. The rest of the equation is just standard LASSO which uses cross-entropy loss and $L1$ regularization. σ is the sigmoid function and $\mathbf{w}$ are the trainable LASSO weights. Note that $L1$ regularization

leads to a sparse weight vector which means the explanation is selective. k refers to the number of perturbed samples and has to be specified by the user.

The original formulation can be adapted to other intrinsically explainable models or sampling techniques. One limitation of LIME is that the weights are not intrinsically explainable when the input features are linearly correlated, which is often the case. SHAP is a technique that mitigates that problem but comes at higher computational costs [22].

2.2.4 Explainability Metrics in NLP

In general, there is no consensus on how to evaluate explainability quantitatively. This is due to the lack of common understanding of a *good explanation*. While natural language explanations are usually evaluated by humans [10], highlight-based techniques can be evaluated more objectively due to their quantitative nature. In this study, we distinguish between two approaches: faithfulness and plausibility [33]. Note that other notions of explainability evaluations exist [76]. For example, consistency [77] assesses how deterministic or implementation invariant the technique is, compactness [78] considers the size of the explanation where less is better and confidence [79] measures the certainty and likelihood of explanations.

Faithfulness measures to what extent the highlights reflect the model's decision process. Generally, faithfulness metrics rely on removing tokens from the input sequence based on the explanation and measuring the change in prediction. While several faithfulness metrics exist, they are "not always consistent with each other and even lead to contradictory conclusions" [33]. The study compared six faithfulness metrics and concluded that comprehensiveness and sufficiency are the most diagnostic and the least complex. For this reason, we evaluate the LIME predictions on these two metrics in the experiments later. Comprehensiveness and sufficiency can be summarised as follows:

- **Comprehensiveness**, proposed by [5], suggests that an explanation is faithful if

the prediction strongly deviates when the most important tokens are removed from the input sequence. More formally,

$$\text{COMP}(\mathbf{x}, c, k) = p(c \mid \mathbf{x}; \theta) - p(c \mid \mathbf{x} \backslash \mathbf{x}_k; \theta) \tag{2.5}$$

Where $p(c \mid \mathbf{x}; \theta)$ denotes the model's prediction for class c on the entire input sequence and $p(c \mid \mathbf{x} \backslash \mathbf{x}_k; \theta)$ denotes the model's prediction when the top k most important tokens $\mathbf{x}_k$ are removed from the input string. Practically, we obtain the most important tokens by taking the top t percent tokens from the list of tokens with associated real-valued importance scores generated by a highlight-based explainability technique. We denote this explanation as $\mathbf{x}_t$. To enhance the metric's reliability, the original paper proposes aggregated comprehensiveness with bins $t \in T = \{1\%, 5\%, 10\%, 20\%, 50\%\}$ [5].

$$\text{COMP}_{\text{agg}}(\mathbf{x}, c) = \sum_{t \in T} \text{COMP}(\mathbf{x}, c, t) = \sum_{t \in T} \left(p(c \mid \mathbf{x}; \theta) - p(c \mid \mathbf{x} \setminus \mathbf{x}_{:t}; \theta) \right) \tag{2.6}$$

- **Sufficiency**, also proposed by [5], is a closely related metric. It measures whether the explanation itself holds sufficient information to obtain the original prediction.

$$\text{SUFF}(\mathbf{x}, c, k) = p(c \mid \mathbf{x}; \theta) - p(c \mid \mathbf{x}_k; \theta) \tag{2.7}$$

Similarly, aggregated sufficiency can be defined as

$$\text{SUFF}_{\text{agg}}(\mathbf{x}, c) = \sum_{t \in T} \text{SUFF}(\mathbf{x}, c, t) = \sum_{t \in T} (p(c \mid \mathbf{x}; \theta) - p(c \mid \mathbf{x}_t; \theta)) \tag{2.8}$$

The explanation is considered faithful if comprehensiveness is high as removing the explanation leads to a less confident prediction. Inversely, the explanation is more faithful if sufficiency is low indicating that the explanation itself contains enough information for the model's original prediction. An illustration of both metrics is depicted in figure 2.9.

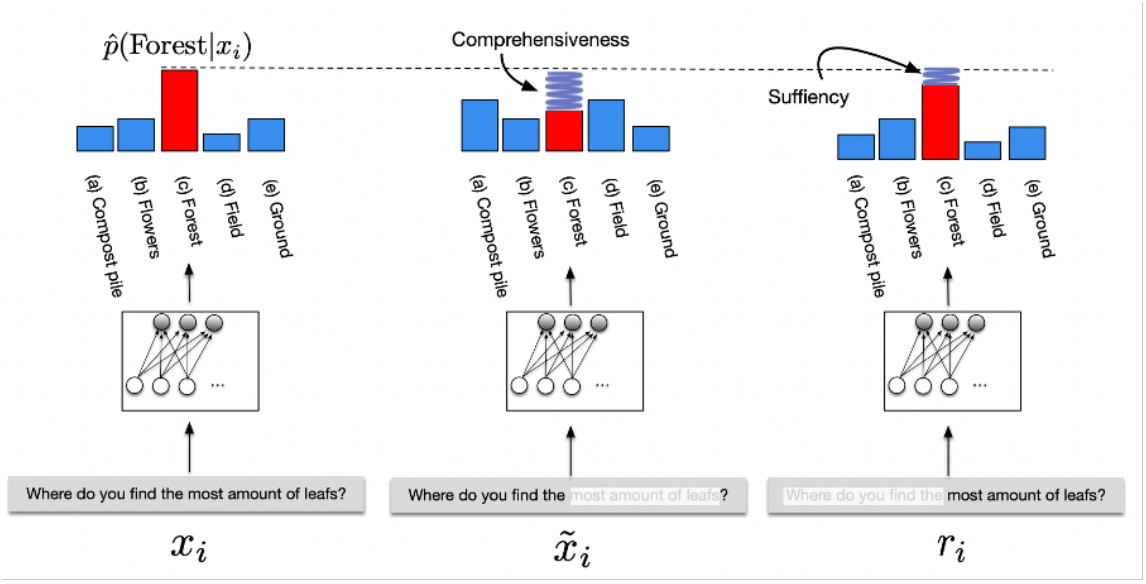


Figure 2.9: Visualistion of comprehensiveness and sufficiency on CoS-e dataset from [5].

Plausibility, as compared to faithfulness, defines a *good explanation* by the intersection between the highlights generated by an explainability technique and some human-annotated highlights. In other words, plausibility measures the "agreement between extracted and human rationales" [5]. Plausibility fundamentally differs from faithfulness in that plausible explanations do not reveal whether the model actually relied on the explanation. Assessing plausibility typically requires human evaluation [80]. However, more recently some existing datasets have been extended by human-annotated highlights [5] which allows for more quantitative evaluation of plausibility. In this thesis, for in-

stance, we use two datasets with human-annotated highlights (CoS-e [81] and e-SNLI [10]). Similarly to [5] we report on intersection over union (IOU) token level f1 scores (TokenF1). Note that originally these metrics were discretised by a threshold to obtain either *good* or *bad* as the metric was used for training a classifier. In this study, we keep the metrics continuous.

- **Intersection over Union** (IOU) measures the overlap between the human explanation and the generated explanation. As proposed by [5], we take the number of most important tokens according to the average explanation length provided by humans for each dataset. Suppose $\mathbf{x}_1$ is the set of tokens from the human explanation and $\mathbf{x}_2$ is the set of generated tokens. Then IOU can be formalised by:

$$\text{IOU}(\mathbf{x}_1, \mathbf{x}_2) = \frac{|\mathbf{x}_1 \cap \mathbf{x}_2|}{|\mathbf{x}_1 \cup \mathbf{x}_2|} \tag{2.9}$$

- **Token level f1 score** (TokenF1) is the harmonic mean between recall and precision and can be defined on a token level as follows

$$\begin{aligned} &\text{tp} = |\mathbf{x}_2 \cap \mathbf{x}_1|\,, \quad \text{fp} = |\mathbf{x}_2 \setminus \mathbf{x}_1|\,, \quad \text{fn} = |\mathbf{x}_1 \setminus \mathbf{x}_2|\,, \\ &\text{precision} = \tfrac{\text{tp}}{\text{tp+fp}}, \quad \text{recall} = \tfrac{\text{tp}}{\text{tp+fn}}, \quad \text{TokenF1} = \tfrac{2\times\text{precision}\times\text{recall}}{\text{precision+recall}} \end{aligned} \tag{2.10}$$

2.3 Explainability and Scaling Laws

So far, we have introduced transformer-based language models and various explainability techniques. In this section, we review literature that combines both fields. We also review literature related to scaling laws of large language models. We were not able to find any studies investigating the relationship between explainability and model size in transformer models but we touch upon the well-established performance interpretability tradeoff.

This thesis is strongly inspired by the work of [5], which introduced both faithfulness metrics, comprehensiveness and sufficiency. The study has applied the metrics on seven different datasets, including two datasets used in the experiments of this thesis (CoS-e, e-SNLI). The authors used a BERT model for a contextualised token representation followed by a bidirectional LSTM layer. Two of seven datasets (BoolQ, Evidence Inference) have long input sequences with more than 1000 tokens. BERT limits the input sequence to 512 tokens, so a GloVe embedding was used instead of BERT. The GloVe models are neglected in this comparison as they are not transformer-based. The study used three post hoc highlight-based explainability techniques: LIME, attention weights and vanilla gradients. Additionally, the metrics were reported for random assignment of importance scores averaged over 10 runs.

Regarding the BERT-based models, LIME achieved the highest average comprehensiveness score (0.254) across all five datasets [5]. On e-SNLI and CoS-e the model achieved 0.437 and 0.223, respectively, with LIME explanations. Attention and gradient-based explanations performed very poorly compared to LIME (0.05 and 0.09, respectively on average). A similar trend holds for sufficiency with LIME achieving the lowest score on average (0.112) Sufficency on e-SNLI and CoS-e were 0.389 and 0.143 respectively. The other techniques also achieve low sufficiency but compared with sufficiency for random explanations attention and gradient performed poorly. The results are summarised in table 2.1.

	↑ Comprehensiveness	↓ Sufficiency
Attention	0.077 ± 0.036	0.214 ± 0.192
Gradient	0.116 ± 0.043	0.202 ± 0.144
LIME	0.254 ± 0.092	0.112 ± 0.157
Random	0.055 ± 0.020	0.248 ± 0.147

Table 2.1: Average aggregated comprehensiveness (high is better) and sufficiency (low is better) across five datasets (Movies, FEVER, MultiRC, CoS-e, e-SNLI) [5].

The datasets from the same study also provided human-annotated highlights. Therefore, plausibility metrics in terms of intersection over union (IOU) and token level f1 scores were reported as well. However, the tested models were specifically trained to predict explanations and are therefore not comparable with the model-agnostic approaches presented in this thesis. Training intrinsically explainable transformer models based on human annotation is a different approach to NLP explainability. An encoder decoder-based baseline model is presented in the same study [5]. Note that the study uses different bins to calculate the faithfulness metrics which makes the results not comparable to our results. Unlike [5] our experiments measure faithfulness with respect to varying model sizes. Additionally, we also analyse the metrics on a label and prediction level.

As discussed in section 2.1.4 pre-trained transformer-based models dominate in most NLP benchmarks. Within the family of transformer-based pre-trained models, we find that "language modelling performance improves smoothly with model size, training dataset size and amount of compute" [6]. Note that model size here refers to the number of parameters in the neural network, excluding embedding. The study empirically shows that there is a "power-law relationship with each individual factor when not bottlenecked by the other two", as depicted in figure 2.10.

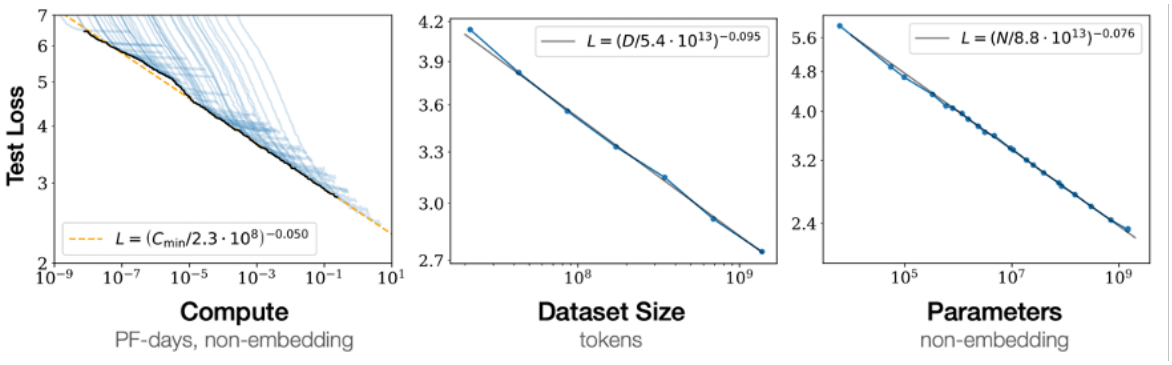


Figure 2.10: Scaling laws of language models [6].

The study also suggests that model performance depends more on scale than on shape (e.g. depth and width in the neural network). The tested models are generative decoder-

only transformers trained on the WebText dataset [55] and tokenized using byte pair encoding [82].

Another study [7] further undermines how the compute requirements have accelerated over the past few years for training advanced ML systems. As illustrated in figure 2.11, the study highlights the emergence of an era of large-scale ML models.

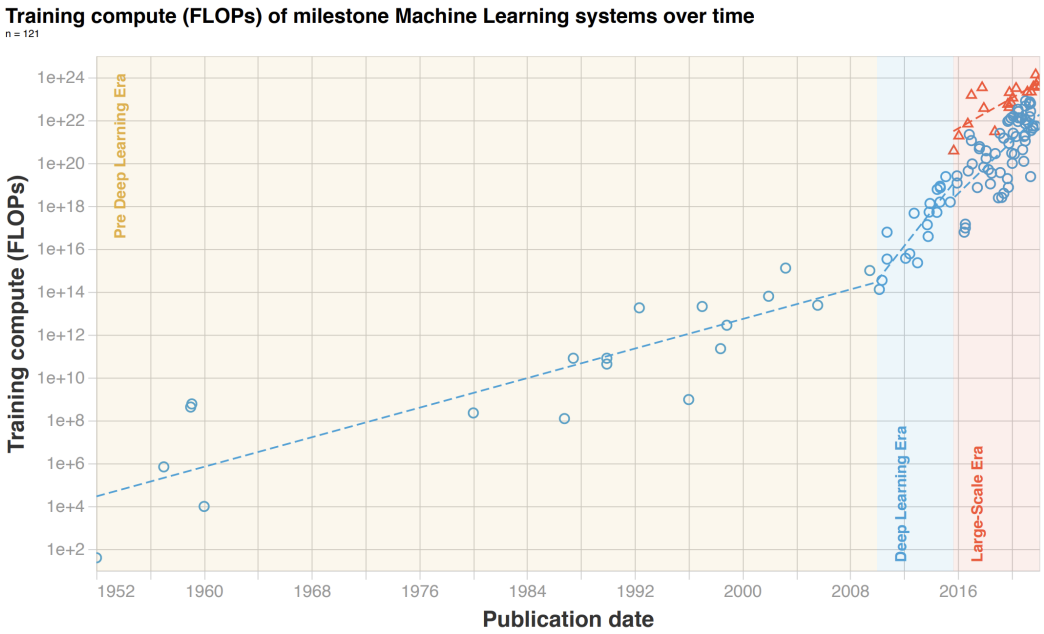


Figure 2.11: ML models between 1952 and 2022 with respect to training compute in FLOPs [7].

Within this new era of large-scale models, we also observe an increase in model size as depicted in 2.12. The trend seems to continue as it is rumoured that GPT-4 has about 100 trillion parameters [21], although no official information is provided from OpenAI.

This project is particularly investigating the interplay between model size and explainability. A well-established concept is the performance vs. interpretability trade-off [9] which states that the opaqueness of the model architecture is directly related to the

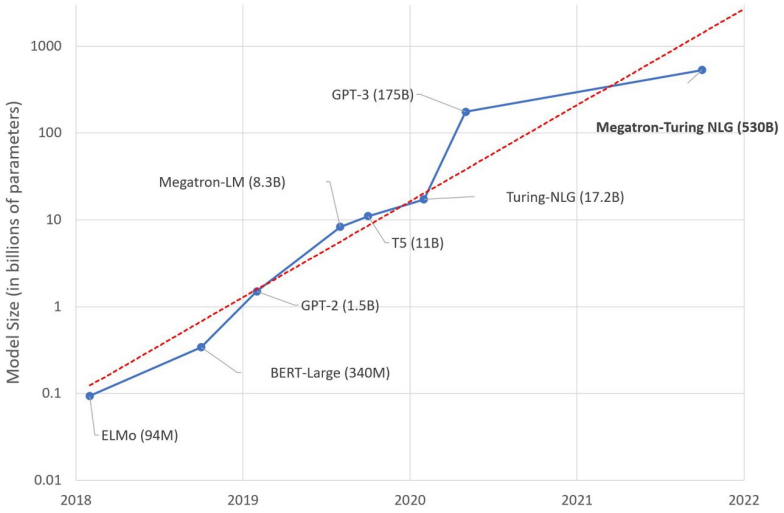


Figure 2.12: Transformers between 2018 and 2022 w.r.t. model size in number of parameters [8].

model's intrinsic explainability (see figure 2.13). However, the figure below relates rather to intrinsic model-specific explainability. To the best of our knowledge, no research has been conducted before regarding posthoc explainablity with respect to model size. The following chapter will introduce the methodology for empirically testing the quality of LIME explanations on BERT-based transformer models of four different sizes.

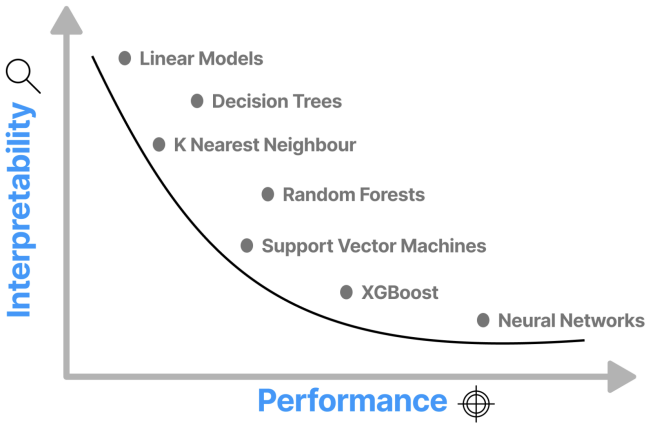


Figure 2.13: Performance vs. interpretability trade-off [9].

Chapter 3

Methodology

This chapter introduces all components used in the experiments, including models, tasks, datasets, explainability technique and metrics. Our experiments use fine-tuned versions of DeBERTaV3 of four different sizes on two tasks, natural language inference and zero shot classification. We use three datasets (MNLI, CoS-e, e-SNLI) and one explainability technique (LIME). We report on standard classification metrics for evaluating the models' performance and use faithfulness and plausibility metrics to evaluate the explanations.

3.1 Models

This section introduces the models used in the experiments. We touch upon architectural details, including model sizes, training procedures and an overview of previous performance benchmarks.

The models used in the experiments are fine-tuned versions of DeBERTaV3 [12], which is a modification of the first version of DeBERTa (Decoding-enhanced BERT with disentangled attention) [52]. DeBERTa, introduced by Microsoft in 2020, uses two novel techniques compared to the original BERT model: disentangled attention and mask-enhanced decoding. Both, the original transformer architecture and BERT, combine the content and position representation in the attention mechanism using one single vector. DeBERTa, on the other hand, represents each token using two vectors to encode position

and content separately. Although this approach considers relative positioning, it fails to address absolute position. For example, consider the sentence: "A new store opened beside the new mall" [52]. Assume during pre-training *store* and *mall* are masked. Using only relative positioning, the model would not be able to differentiate between the two as both follow the word *new*. While BERT integrates absolute positions in the input layer, DeBERTa integrates them after all the transformer layers and before the softmax layer. The authors suggest that "early incorporation of absolute positions used by BERT might undesirably hamper the model from learning sufficient information of relative positions" [52].

Similar to BERT, DeBERTa was trained on BookCorpus [49] and English Wikipedia. DeBERTa uses three additional datasets [52], resulting in a total of 78GB of training data. Analogue to BERT, DeBERTa was introduced in a base configuration (12 layers, 768 hidden size, 12 heads) and a large configuration (24 layers, 1024 hidden size, 16 heads).

DeBERTaV3 [12], introduced in 2023 by Microsoft, improves upon DeBERTa by replacing the masked language modelling (MLM) task for pre-training by replaced token detection (RTD), which was first proposed in the ELECTRA model [83]. Instead of masking the input tokens and then predicting the original tokens, RTD corrupts input tokens by replacing them with alternatives sampled from a generator network. The task is then to train a discriminator network that predicts whether or not the tokens were corrupted by the generator. This technique has been shown to be more compute efficient than MLM, especially for smaller networks. Compared to the first version DeBERTaV3 uses an additional dataset (CC-News), which results in 160GB of training data. In addition to a base and large model, DeBERTaV3 comes in small and xsmall versions with fewer parameters. An overview of all models' architecture and performance benchmarks is summarised in table 3.1. Performance was reported on the GLUE benchmark [47], the (matched) MNLI dataset [11], and the SQuAD 2.0 dataset [84].

	Parameters (in millions)	Layers	Hidden Size	Attention Heads	MNLI (accuracy)	SQuAD 2.0 (F1)	GLUE (avg. score)
large	304	24	1024	12	91.8	91.5	91.37
base	86	12	768	12	90.6	88.4	-
small	44	6	768	12	88.2	82.9	-
xsmall	22	12	384	6	88.1	84.8	-

Table 3.1: Performance and architecture comparison for DeBERTaV3 models [12].

Note that the models used in the experiments are fine-tuned versions of DeBERTaV3 on the downstream task of natural language inference. The models were fine-tuned on the SentenceBERT framework [85] using the MNLI [11] and SNLI [13] datasets and are publically available on Huggingface [34].

3.2 Tasks

In the following, we discuss the tasks considered in the experiments, i.e. natural language inference and zero shot classification.

3.2.1 Natural Language Inference

Natural language inference (NLI) is a standard text classification task that identifies logical relationships between pairs of sentences. The task is to determine whether a hypothesis is true (*entailment*), false (*contradiction*) or undetermined (*neutral*) given a premise. Examples from the SNLI [13] dataset are depicted in table 3.2

Premise	**Hypothesis**	**Label**
Some dogs are running on a deserted beach.	There are multiple dogs present.	entailment
A man playing an electric guitar on stage.	A man playing banjo on the floor.	contradiction
A boy hits a tennis ball on a court.	The boy is good at tennis.	neutral

Table 3.2: NLI examples from the SNLI dataset [13].

3.2.2 Zero Shot Classification

Zero shot classification is another text classification task where the model predicts from a set of labels that the model has not been trained on. Examples are depicted in table 3.3.

Sentence	Candidate Labels	True Label
Cristiano Ronaldo is the best player in the world.	[Politics, Science, Football]	Football
The iPhone 15 will be released in September.	[Technology, Sports, Weather]	Technology
Berlin has really good techno clubs.	[Agriculture, Education, Music]	Music

Table 3.3: Examples for zero shot classification.

Note that Huggingface utilizes a method proposed by [86] which allows any NLI fine-tuned model to perform zero-shot classification. The idea is to simply take the sentence as premise and construct a hypothesis as: *this text is about ...* followed by a candidate label. The probability for entailment reflects the model's certainty for predicting the label. The same step is repeated for all possible labels. Passing the probabilities for entailment through a softmax layer results in the final prediction. This way we do not have to change the DeBERTaV3 models and can use the same models for both tasks. However, this translation is computationally more expensive as each candidate label requires its own forward pass.

3.3 Datasets

The experiments are conducted on three datasets reflecting the two tasks introduced above. We use MNLI [11] and e-SNLI [10] for NLI and CoS-e [14] for zero shot classification. Two datasets (e-SNLI, CoS-e) contain human-annotated highlights indicating important tokens. We use these tokens to compute the plausibility metrics. During the explanation evaluation, we dropped 503 instances from CoS-e as the ground truth highlights contained all input tokens instead of a subset. 12 instances from e-SNLI were

dropped because the highlight indices didn't match the marked tokens. As the focus of the thesis is to evaluate the performance and explainability of language models, we only use the validation sets and omit training sets. The NLI datasets are balanced as shown in figure 3.1.

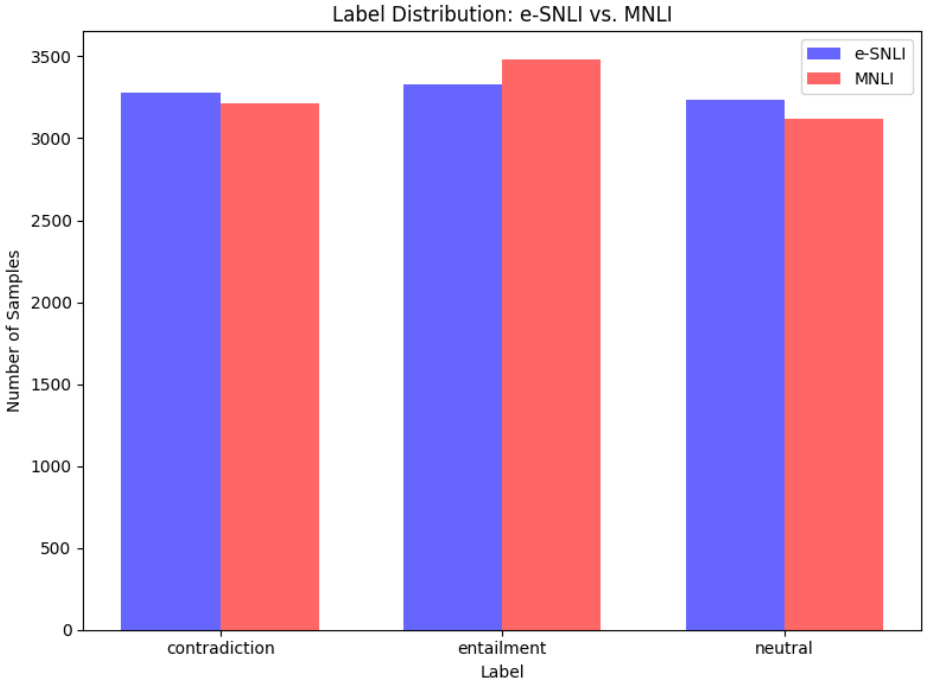


Figure 3.1: Label distribution in e-SNLI [10] and MNLI [11] validation sets.

An overview of the key characteristics of all datasets including average number of input tokens and average explanation input ratios is depicted in table 3.4. The ratio is calculated by dividing the number of highlighted tokens by the number of input tokens and taking the average over all validation instances excluding low-quality explanations.

	Train	**Validation**	**Test**	**Mean Token Count** (± std)	**Explanation-Input-Ratio** (± std)
MNLI	392702	9815	10000	29.18 ± 16.26	-
e-SNLI	549367	9842	9824	21.45 ± 7.52	0.1985 (±0.193)
CoS-e	9741	1221	-	13.13 ± 5.23	0.2605 (±0.137)

Table 3.4: Exploratory data analysis.

3.3.1 MNLI

MNLI [11], or MultiNLI, is a crowdsourced dataset for natural language processing. It differs from the previously introduced SNLI [13] dataset by including a range of distinct genres from both written and spoken English text. MNLI offers two different validation sets, a matched and a mismatched version. The matched version contains the same genres as in the training set. The mismatched version contains genres that do not appear in the training set, which allows developers to test the model's generalizability across different genres. For the experiments, we chose to use the matched validation set retrieved from Huggingface [87]. Examples from the MNLI dataset are illustrated in table 3.5

Premise	**Hypothesis**	**Genre**	**Label**
Look, there's a legend here.	See, there is a well-known hero here.	Fiction	Entailment
Yeah, I know, and I did that all through college and it worked too.	I did that all through college but it never worked.	Telephone	Contradiction
Boats in daily use lie within feet of the fashionable bars and restaurants.	Bars and restaurants are interesting places.	Travel	Neutral

Table 3.5: MNLI [11] examples.

3.3.2 e-SNLI

e-SNLI [10] is an extended version of the Standford Natural Language Inference (SNLI) [13] dataset. For the premises SNLI uses image captions from the Flickr30k corpus [88]. Then Amazon Mechanical Turk workers were asked to provide hypotheses for each label which resulted in a balanced dataset. SNLI was introduced before MNLI and is considered less diverse in topic and style as it is just based on image captions.

In e-SNLI human annotators extended the dataset by highlighting tokens in the premise and/or hypothesis that they considered important. Apart from the highlights the dataset was additionally extended by natural language explanations in free text form. The anno-

tations involved 6325 Amazon Mechanical Turk workers with 86 explanations on average.

Dataset examples are depicted in table 3.6. Note that in the experiments we only use the highlight-based explanations and omit the natural language explanations. We retrieve the dataset directly from the official GitHub repository [89].

Premise	**Hypothesis**	**Label**	**Explanation**
An adult dressed in black holds a stick .	An adult is walking away, empty-handed .	Contradiction	Holds a stick implies using hands so it is not empty-handed.
A child in a yellow plastic safety swing is laughing as a dark-haired woman in pink and coral pants stands behind her.	A young mother is playing with her daughter in a swing.	Neutral	Child does not imply daughter and woman does not imply mother.
A man in an orange vest leans over a pickup truck .	A man is touching a truck.	Entailment	Man leans over a pickup truck implying that he is touching it.

Table 3.6: e-SNLI [10] examples.

3.3.3 CoS-e

Similar to e-SNLI, the CommonSense explanation (CoS-e) dataset is a dataset extended by human-annotated highlights and natural language explanations. CoS-e extends the CommonSenseQA (CQA) dataset [90] which is a multiple-choice question-answering dataset. Each question contains five different candidate labels, and the labels differ for each question. We, therefore, use CoS-e for zero shot classification.

Examples are depicted in table 3.7. Again, we omit the natural language explanations and only use the highlights to evaluate the models' plausibility. The dataset was retrieved from Huggingface [91].

Question	Candidate Labels	Label	Explanation
He was a sloppy eater, so where did he leave a mess?	sailboat, desk, closet, table, apartment	table	One normally eats at a table.
Where can someone get a new saw?	hardware store, toolbox, logging camp, tool kit, auger	hardware store	Hardware stores sell new saws.
Many homes in this country are built around a courtyard. Where is it?	hospital, park, spain, office complex, office	spain	Spain is the name of a country.

Table 3.7: CoS-e [14] examples.

3.4 Explainability Technique

Our experiments use Local Interpretable Model Agnostic Explanations (LIME) [4]. LIME was introduced in detail in section 2.2.3. As described in the explainability and scalability section 2.3, LIME has shown to be the most faithful technique among LIME, attention weights and gradient-based explanations. Furthermore, attention weights are considered to be unreliable explanations [74] and gradient-based methods can be manipulated [92] which further justifies LIME. We evaluate the LIME explanations on their faithfulness and plausibility while varying the model size. We use the text module from the original LIME package [93]. Implementation details will be discussed in chapter 4.

3.5 Metrics

The experiments evaluate both the models' performance on the tasks introduced in section 3.2 and on the LIME explanations. The explanations are evaluated on their faithfulness and their plausibility. Both concepts were described in section 2.2.4.

3.5.1 Performance

As far as model performance is concerned, we report on the standard classification metrics accuracy, precision, recall and f1 scores for the NLI task. We can safely use macro-averaged scores as the NLI datasets as classes are well balanced (see figure 3.1). Since CoS-e has different labels for each question, we only report accuracy as the other metrics do not apply in a zero shot classification setting. The metrics can be summarised as follows.

- **Accuracy** represents the fraction of correct predictions out of all predictions.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{3.1}$$

- **Precision** for a specific class is the ratio of correctly predicted observations of this class over the total number of predictions for this class.

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}, c \in \text{Classes} \tag{3.2}$$

  where TP_c are the true positives for a specific class, FP_c are the false positives for a specific class , and Classes = {contradiction, entailment, neutral}

- **Recall** for some class is the ratio of correctly predicted positive observations to all observations in that class

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}, \ c \in \text{Classes} \tag{3.3}$$

  where FN_c are the false negatives for a specific class.

- **F1 score** for some class is the harmonic mean between precision and recall and therefore expresses the balances between the two.

$$\mathrm{F1_c} = 2 \times \frac{\mathrm{Precision_c} \times \mathrm{Recall_c}}{\mathrm{Precision_c} + \mathrm{Recall_c}}, \ \mathrm{c} \in \mathrm{Classes} \tag{3.4}$$

- **Macro-Averaged Scores** compute the metric independently for each class and then take the average.

$$\text{Macro-Precision} = \frac{1}{3} \sum_{\mathrm{c} \in \mathrm{Classes}} \mathrm{Precision}_c \tag{3.5}$$

$$\text{Macro-Recall} = \frac{1}{3} \sum_{\mathrm{c} \in \mathrm{Classes}} \mathrm{Recall}_c \tag{3.6}$$

$$\text{Macro-F1} = \frac{1}{3} \sum_{\mathrm{c} \in \mathrm{Classes}} \mathrm{F1}_c \tag{3.7}$$

3.5.2 Faithfulness

Faithfulness aims to measure the extent to which an explanation reflects the model's decision process. We use comprehensiveness and sufficiency as described in section 2.2.4 to evaluate the models' faithfulness. When reporting on a global level, we take the average aggregated comprehensiveness and sufficiency respectively.

3.5.3 Plausibility

Plausibility captures the agreement between a human explanation and an explanation generated by some explainability method. More specifically, we report on the intersection over union (IOU) and token level f1 scores (TokenF1) as introduced in section 2.2.4. Again, averages of these measures are taken for global evaluation of plausibility.

3.6 Chapter Summary

In summary, our experiments use fine-tuned DeBERTaV3 models of four different sizes. The models are applied on two tasks, natural language inference (MNLI, e-SNLI) and zero shot classification (CoS-e). To explain the models' predictions we apply LIME. The explanations are evaluated based on their faithfulness (comprehensiveness and sufficiency) and their plausibility (IOU and TokenF1).

Chapter 4

Experiments and Results

This chapter summarises the experiments and results conducted in this project. We first evaluate the models on all three datasets and find an increase in performance for larger models. Then we evaluate the LIME explanations for all models by applying faithfulness and plausibility metrics. First, the metrics are applied to local instances to build an intuition. Finally, the explanations are evaluated more globally. We explain 100 instances from the validation set of each dataset. We empirically conclude that in zero shot classification the LIME explanations are more comprehensive as the model size increases while plausibility remains constant. We further break down the results by label and by correct or incorrect predictions The results are accompanied by tables and figures throughout.

4.1 Model Evaluation

The four DeBERTaV3 models were evaluated on the validation sets of all three datasets (MNLI, e-SNLI, CoS-e). We use Google Colab for computational resources (Nvidia's T4 GPU, 51GB RAM). The models were evaluated on standard classification metrics (accuracy, precision, recall, f1). To capture the robustness of these metrics, 95% confidence intervals were calculated using the bootstrapping method with 1000 iterations [94].

The performance scores are visualised in figure 4.1 and table 4.1. Since the dataset classes are balanced (see figure 3.1) all classification metrics are very similar. Therefore

we report on accuracy only here. Macro weighted f1 scores, recall and precision are laid out in appendix A.

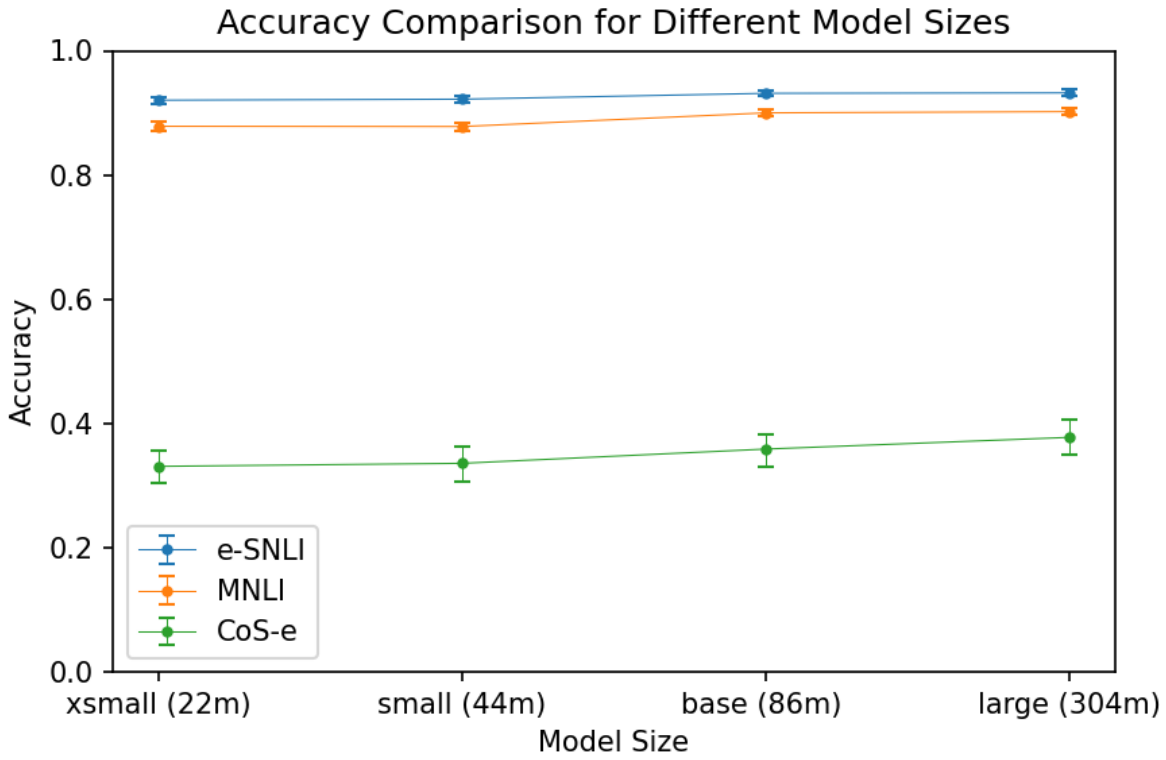


Figure 4.1: Performance comparison between datasets across different model sizes. The error bars indicate 95% confidence intervals.

We find that across all three datasets, the performance increases with model size. While the accuracy in the NLI tasks improves by 1.2% (e-SNLI) and 2.4% (MNLI) the accuracy in zero shot classification improves more significantly by 4.7% (CoS-e). As all models were fine-tuned on MNLI and SNLI high performance on the NLI tasks is expected. The performance on CoS-e is relatively low but note that every question in this dataset has five possible classes as opposed to three classes in NLI. Since the focus of this project is explainability evaluation we did not further fine-tune the models on CoS-e.

Dataset	Model Size	Accuracy	95% C.I.
MNLI	xsmall	0.878	(0.871, 0.885)
	small	0.878	(0.872, 0.884)
	base	0.900	(0.894, 0.906)
	large	**0.902**	(0.896, 0.908)
e-SNLI	xsmall	0.920	(0.915, 0.925)
	small	0.922	(0.917, 0.927)
	base	0.931	(0.926, 0.936)
	large	**0.932**	(0.927, 0.937)
CoS-e	xsmall	0.331	(0.305, 0.355)
	small	0.336	(0.306, 0.362)
	base	0.359	(0.330, 0.383)
	large	**0.378**	(0.349, 0.406)

Table 4.1: Model accuracy with 95% confidence intervals.

4.2 Explainability Evaluation

In this section, the LIME explanations are evaluated based on their faithfulness and plausibility. While faithfulness describes the extent to which the explanation reflects the models' decision, plausibility captures the agreement between a generated explanation and a human explanation. Faithfulness is measured in terms of comprehensiveness and sufficiency. Plausibility is calculated by the intersection over union (IOU) and the token level f1 score (TokenF1). All metrics were introduced and formally defined in section 2.2.4. Note that evaluating plausibility requires ground truth highlight-based explanations. CoS-e and e-SNLI both provide human-annotated highlights. MNLI on the other hand does not contain ground truth explanations. The explanations on MNLI are therefore only evaluated on their faithfulness.

Regarding the experimental setup, we use the standard LIME package [93]. The explanations were calculated based on 100 perturbed samples with cosine distance to the input sequence. The input is tokenized by whitespace. The explanations were computed on

the same Google Colab setup (Nvidia's T4 GPU, 51GB RAM).

The following section aims to provide an intuition for LIME explanations and the explainability metrics for faithfulness and plausibility. Thereafter, the number of explanations is scaled up for each model on all three datasets in order to draw quantitative conclusions about the quality of explanations with respect to model size. Due to the computation intensity of LIME, we restrict the number of explanations to 100.

4.2.1 Local Evaluation

We discuss two local instances, one from the CoS-e dataset and one from the e-SNLI dataset. For prediction, we use the xsmall DeBERTaV3 model.

Example 1: Consider the following instance from the CoS-e dataset.

Question	**Candidate Labels**	**Label**
Bob the lizard lives in a warm place with lots of water. Where does he probably live?	rock, tropical rainforest, jazz club, new mexico, rocky places	tropical rainforest

Table 4.2: Local instance in CoS-e validation set.

Predicting this instance on the model yields probability scores for each candidate label. The model correctly identifies *tropical rainforest* as the most likely answer (0.53). Applying LIME on this instance with respect to the predicted label provides the following highlights as illustrated in figure 4.2. Although LIME maps a score for each input token, the figure only displays a subset of token scores. Note that throughout the experiments we apply LIME with respect to the predicted label not necessarily to the true label.

LIME highlights the token *water* as the most important token for the model's prediction. Note that token scores can also have negative scores indicating that a token has a negative effect on a particular class prediction.

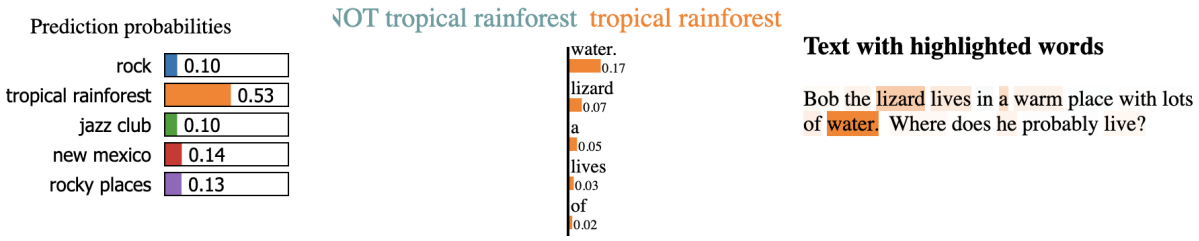


Figure 4.2: LIME example on a CoS-e instance using the xsmall DeBERTaV3 model.

To measure the quality of the LIME explanation we evaluate its faithfulness and plausibility. Faithfulness is measured by the comprehensiveness and sufficiency metrics. Comprehensiveness measures how much the prediction changes when the explanation (i.e. the most important tokens) is removed from the input sequence. Conversely, sufficiency captures the change in prediction when only the explanation itself is used for prediction. To obtain the most important tokens we take the top 10% of tokens with the most positive impact (i.e. *water*, *lizard*). Now, the model predicts the modified input sequences, that is, the input without explanation for comprehensiveness and the explanation only for sufficiency as summarised in table 4.3.

Full Input	**Input without Explanation**	**Explanation only**
Bob the lizard lives in a warm place with lots of water. Where does he probably live?	Bob the ~~lizard~~ lives in a warm place with lots of ~~water~~. Where does he probably live?	~~Bob the~~ lizard ~~lives in a warm place with lots of~~ water. ~~Where does he probably live?~~

Table 4.3: Modified CoS-e input sequences to calculate comprehensiveness and sufficiency.

The comprehensiveness metric is then calculated as the difference between the probability for the predicted class on the full input and the probability for the predicted class on the input without the explanation. Similarly, sufficiency is the difference in prediction between the full input and the explanation only. The metrics for the CoS-e instance are visualised in figure 4.3. We can see that the comprehensiveness (0.44) is similarly high as the original prediction (0.53) for *tropical rainforest*. Regarding sufficiency (0.3) we observe that using only the explanation the probability for *tropical rainforest* decreases to 0.23

and prediction changes to *rocky places*. We conclude that for this instance the explanation is comprehensive because the model relied on the explanation but not sufficient as the explanation itself is insufficient to retain the prediction.

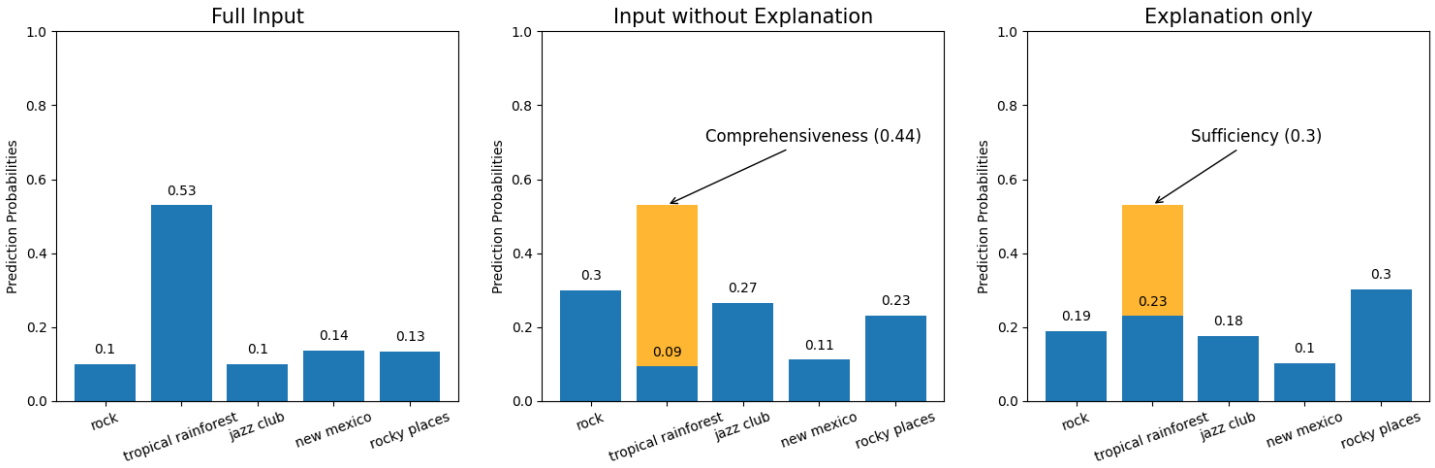


Figure 4.3: Comprehensiveness and sufficiency visualisation on a CoS-e instance.

The metrics therefore capture how faithful the explanation is to the model's internal decision process. The explanation is considered faithful if comprehensiveness is high as removing the explanation leads to a less confident prediction. Inversely, the explanation is more faithful if sufficiency is low indicating that the explanation itself contains enough information for the model's original prediction.

Note that both metrics can become negative as well. Negative comprehensiveness would mean that the model is more confident in its prediction when the explanation is removed which suggests that the explanation is not faithful. Negative sufficiency means the model is more confident in its prediction using only the explanation indicating that the other input tokens are distracting the model. As suggested by [5] sufficiency and comprehensiveness are aggregated over different explanation sizes to take the variable length of explanations into account (see equations in section 2.2.4). In the global evaluation experiment, we aggregate over the top 10%, 30% and 50% most important tokens and report on the average scores.

To measure the plausibility of the LIME-generated explanations we use human-annotated highlights which are available in the e-SNLI and CoS-e dataset. For example, the ground truth explanation of the CoS-e instance above is *lots, of, water* as highlighted in table 4.2. As described in section 2.2.4 plausibility is measured by the intersection of union (IOU) and the token level f1 score (TokenF1). The amount of tokens for the LIME explanation is determined by the average input-explanation-ratio of ground truth explanations which is 26.05% for e-SNLI and 19.85% for CoS-e (see table 3.4).

Following equation 2.9 for IOU and 2.10 for TokenF1 the plausibility of the CoS-e instance results in an IOU score of 0.33 and TokenF1 of 0.5 (see table 4.4). We conclude that the LIME explanation agrees partially with the human explanation for this instance.

Ground Truth Explanation	**LIME Explanation**	**IOU**	**TokenF1**
{lots, of, water}	{water, lizard, a, lives, of}	0.33	0.5

Table 4.4: IOU and TokenF1 on a CoS-e instance.

Example 2: We use the same approach for the NLI datasets. In this second example consider the following instance from the e-SNLI dataset.

Premise	**Hypotensis**	**Candidate Labels**	**Label**
A crowd of people are standing against a railing.	The people are all sitting.	contradiction, entailment, neutral	contradiction

Table 4.5: Local instance in e-SNLI validation set.

Figure 4.4 illustrates the highlighted tokens by LIME on the e-SNLI instance. LIME correctly identifies the token *sitting* to be important for predicting the label *contradiction*. The SEP token in the NLI task is used to indicate the separation between premise and hypothesis which is predefined by the Huggingface API.

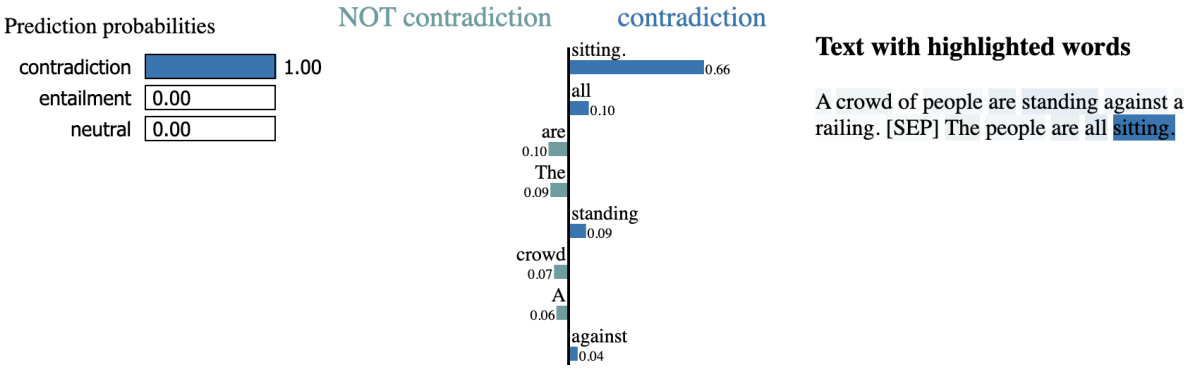


Figure 4.4: LIME example on an e-SNLI instance using the xsmall DeBERTaV3 model.

In this example, we use the top 30% of LIME tokens to obtain the explanation (i.e. *sitting, all, standing*). The modified input sequences for calculating comprehensiveness and sufficiency are displayed in shown in table 4.6.

Full Input	**Input without Explanation**	**Explanation only**
A crowd of people are standing against a railing. [SEP] The people are all sitting.	A crowd of people are ~~standing~~ against a railing. [SEP] The people are ~~all sitting~~.	~~A crowd of people are~~ standing ~~against a railing.~~ [SEP] ~~The people are~~ all sitting.

Table 4.6: Modified e-SNLI input sequences to calculate comprehensiveness and sufficiency.

Predicting on the modified inputs with respect to the predicted class (*contradiction*) yields high comprehensiveness (0.898) and low sufficiency (0.003) which suggests that the LIME explanation is faithful and reflects the model decision process well. Both metrics for this instance are illustrated in figure 4.5.

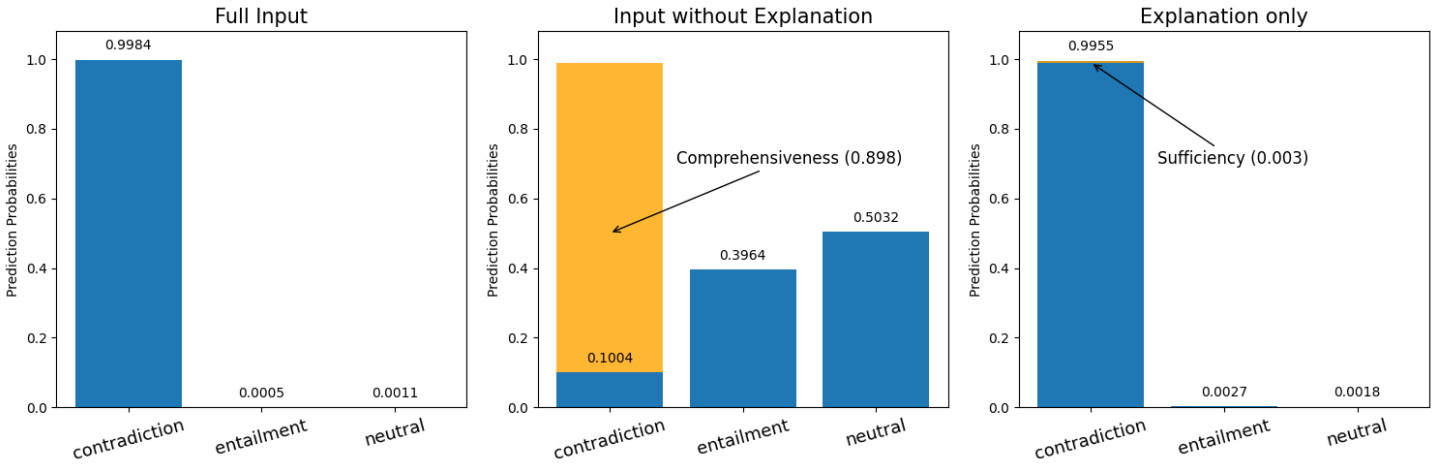


Figure 4.5: Comprehensiveness and sufficiency visualisation on an e-SNLI instance.

To measure the explanation's plausibility we compare the LIME explanation with the human-annotated ground truth explanation. The average ground truth explanation-input-ratio for e-SNLI is 26.05 % which results in the same explanation as before (*sitting, all, standing*). Comparing this LIME explanation with the ground truth explanation (*sitting, standing*) yields the IOU and TokenF1 scores as shown in table 4.7. We conclude that for this instance the model is well explainable using LIME from a faithfulness and plausibility perspective.

Ground Truth Explanation	**LIME Explanation**	**IOU**	**TokenF1**
{sitting, standing}	{sitting, all, standing}	0.66	0.8

Table 4.7: IOU and TokenF1 on an e-SNLI instance.

In this section, we have shown how to systematically evaluate LIME explanation on a local level. The following experiment evaluates the models more globally on 100 instances for each dataset in order to draw more general conclusions about the models' explainability.

4.2.2 Global Evaluation

In this section, we evaluate 100 randomly selected LIME instances on the DeBERTaV3 models of different sizes across the validation sets for all three datasets. While the faithfulness metrics comprehensiveness and sufficiency are reported for all three datasets the plausibility metrics IOU and TokenF1 are reported on e-SNLI and CoS-e only as no ground truth highlights are available in MNLI. We dropped 503 instances from CoS-e as the ground truth highlights contained all input tokens instead of a subset. 12 instances from e-SNLI were dropped because the highlight indices didn't match the marked tokens. All metrics are macro-weighted as classes are well-balanced and mean standard errors are included to capture the robustness of the metrics. The results are summarised in table 4.8 and visualised in figure 4.6.

Dataset	Model Size	Faithfulness		Plausibility	
		↑ Comprehensiveness	↓ Sufficiency	↑ IOU	↑ Token F1
MNLI	xsmall	0.785 (± 0.022)	0.16 (± 0.024)	–	–
	small	0.817 (± 0.022)	**0.131** (± 0.021)	–	–
	base	0.796 (± 0.027)	0.205 (± 0.025)	–	–
	large	**0.823** (± 0.027)	0.19 (± 0.024)	–	–
e-SNLI	xsmall	0.726 (± 0.022)	**0.146** (± 0.018)	**0.282** (± 0.017)	**0.413** (± 0.021)
	small	0.724 (± 0.026)	0.201 (± 0.024)	0.259 (± 0.016)	0.385 (± 0.02)
	base	0.764 (± 0.025)	0.187 (± 0.022)	0.254 (± 0.016)	0.38 (± 0.02)
	large	**0.778** (± 0.025)	0.196 (± 0.023)	0.256 (± 0.017)	0.379 (± 0.021)
CoS-e	xsmall	0.304 (± 0.018)	-0.107 (± 0.014)	0.233 (± 0.013)	0.357 (± 0.019)
	small	0.316 (± 0.019)	**-0.143** (± 0.02)	0.231 (± 0.014)	0.352 (± 0.02)
	base	0.356 (± 0.02)	-0.059 (± 0.019)	**0.235** (± 0.012)	**0.365** (± 0.017)
	large	**0.391** (± 0.022)	-0.08 (± 0.021)	0.23 (± 0.012)	0.358 (± 0.017)

Table 4.8: Macro scores for faithfulness and plausibility including standard errors.

We find that the comprehensiveness of the LIME explanations is increasing with the size of the models across all three datasets. Regarding sufficiency, no clear trend with respect to model size can be observed. Sufficiency is lower (better) for CoS-e than for the NLI tasks suggesting that the models are more confident in their predictions when only

using the explanation. As for plausibility, both metrics (IOU and TokenF1) stay almost constant when the model size increases with slightly higher plausiblity for e-SNLI than for CoS-e.

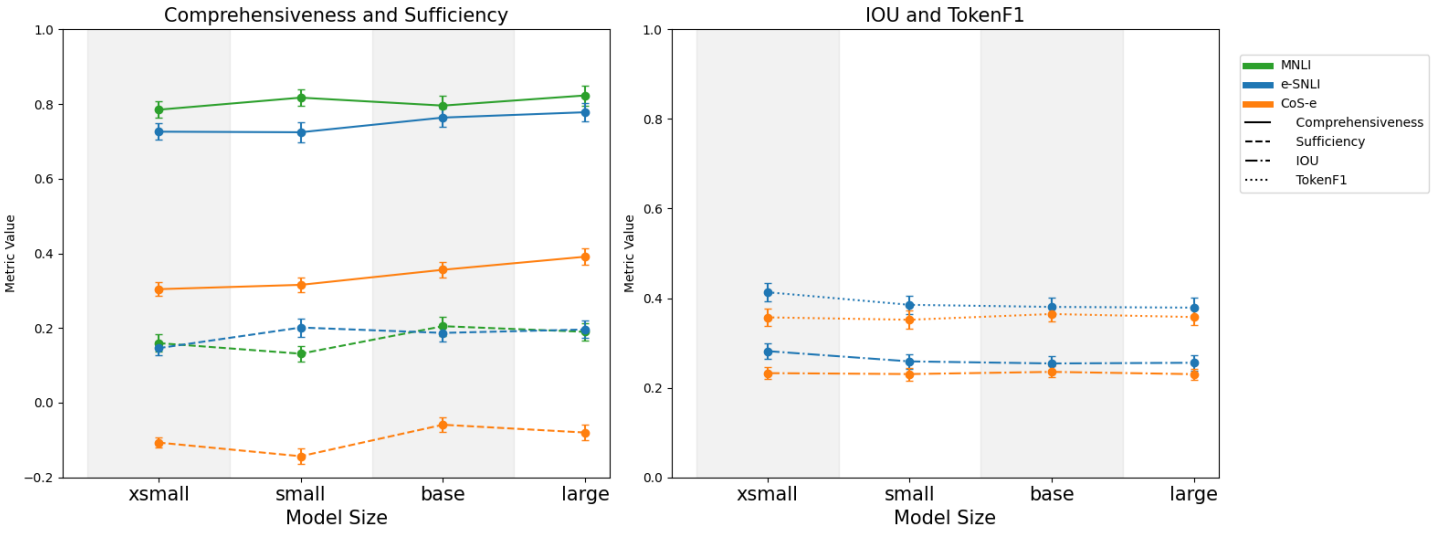


Figure 4.6: Macro faithfulness and plausibility across datasets and model sizes.

To get further insights into the models' explainability we split the metrics by prediction (correct or incorrect) for all three datasets and by label (contradiction, entailment and neutral) for the NLI dataset. Table A.4 and A.5 in the appendix contains a summary of the number of observations for each category for all datasets. Note that while the NLI classes are well balanced, the number of correctly predicted instances overweighs the number of incorrect predictions for the NLI datasets as all models' accuracy is very high (>0.87). This strongly limits the reliability of the metrics for the incorrect instances in the NLI datasets. The reliability is captured by mean standard error bars in the plots. The data points in the plots are slightly misaligned to increase readability and do not imply any shift in model size. The x axis discrete and not real which is underpinned by the shadowed background.

Figure 4.7 independently depicts the faithfulness metrics by label. We observe that for both NLI datasets, contradiction instances tend to achieve higher comprehensiveness than

the other labels. Intuitively, it is easier for the model to explain contradictive pairs of sentences than entailed or neutral pairs. While the models' comprehensiveness improves upon the contradictive pairs in MNLI they improve on the entailment and neutral pairs for e-SNLI.

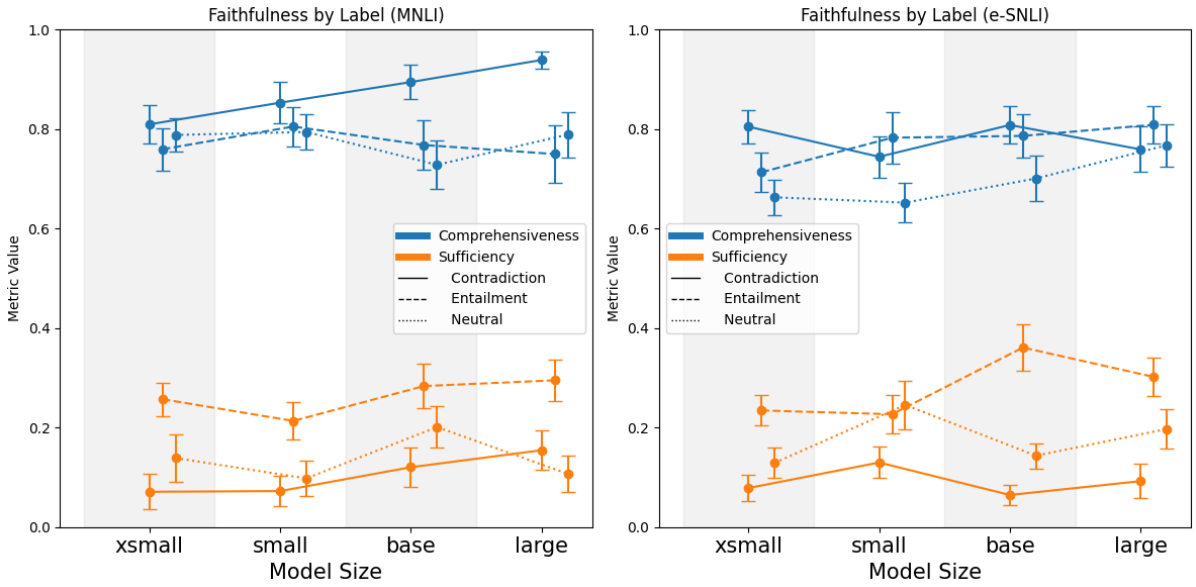


Figure 4.7: Faithfulness by labels for MNLI (left) and e-SNLI (right).

For both datasets, sufficiency tends to be lower for contradiction instances than for entailment and neutral instances. Intuitively, when the LIME explanation contains contradictive words it is likely to predict contradiction for the explanation only with similarly high probability compared to the original input in general. For sufficiency, there's no overall trend observable with respect to model size.

Regarding plausibility by label (figure 4.8), which could only be tested on e-SNLI, we find that both IOU and TokenF1 are uncorrelated to model size. While both metrics are similar for contradiction and entailment pairs the plausibility for neutral pairs is significantly lower across all model sizes for both metrics.

Lastly, we investigated comprehensiveness and plausibility with respect to the model

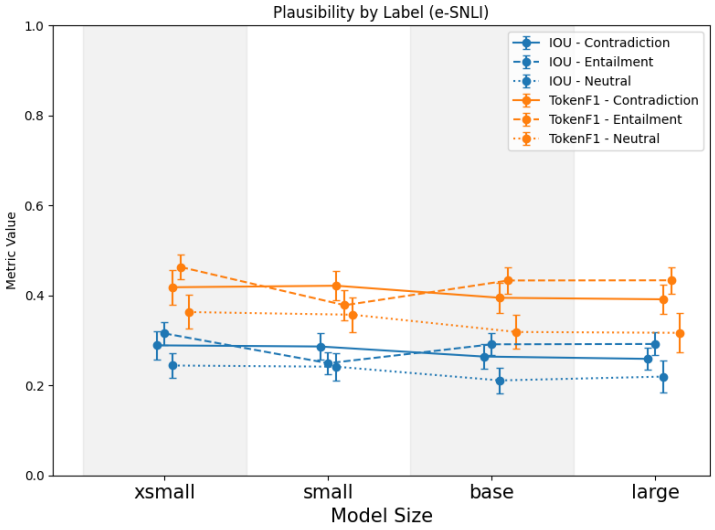


Figure 4.8: Pausibility by label for e-SNLI.

prediction. In figure 4.9 we observe that while comprehensiveness stays almost constant for correct predictions in the MNLI dataset, the comprehensiveness of wrong predictions increases with the model size reaching 0.925 for the large model. Similarly for e-SNLI the comprehensiveness of wrong predictions peaks for the large model at 0.852. In the zero shot classification setting comprehensiveness improves with model size monotonically regardless of the prediction.

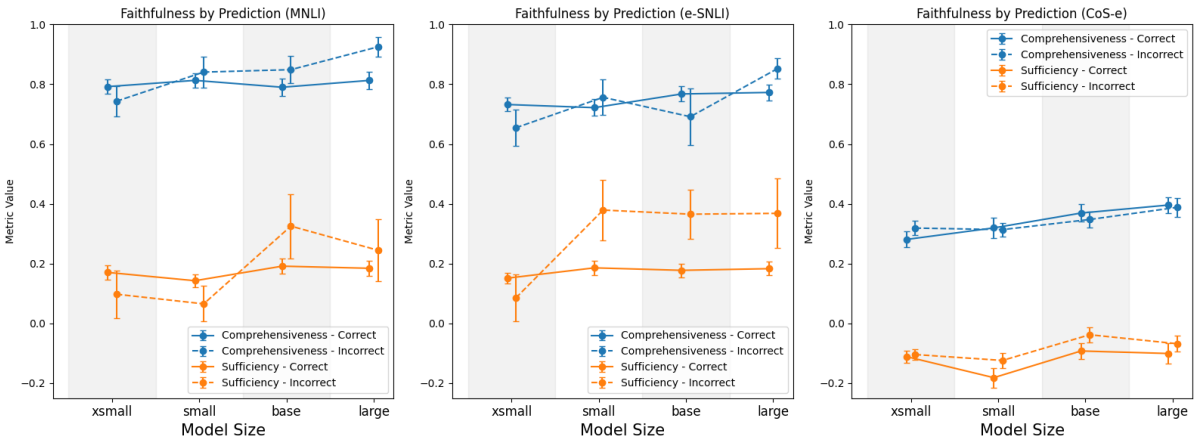


Figure 4.9: Faithfulness by prediction for MNLI (left) and e-SNLI (center) and CoS-e (right).

As for sufficiency, we find that in the NLI task, the larger model tends to have higher sufficiency for wrong predictions than the smaller models while correct predictions have a nearly constant sufficiency of around 0.2. However, as indicated by the error bars, the metrics for incorrect predictions are highly unstable due to the small number of incorrect predictions in the NLI datasets. For CoS-e the models' sufficiency behaves similarly low for both correct and incorrect predictions.

The plausibility metrics for the correct predictions stay almost constant across all model sizes as illustrated in figure 4.10. For e-SNLI, correct predictions have significantly higher plausibility than incorrect predictions which suggests that the human-annotated labels agree better with the LIME explanations for correct predictions than for incorrect predictions. Again, the metrics for wrong predictions are unreliable due to the small sample size which also likely explains the strong fluctuations on the base and large model. We would expect a similar pattern for CoS-e, however, the figure shows that the metrics are not linked to the predictions' correctness which suggests that either the LIME explanations or the human-annotated highlights in CoS-e are of bad quality.

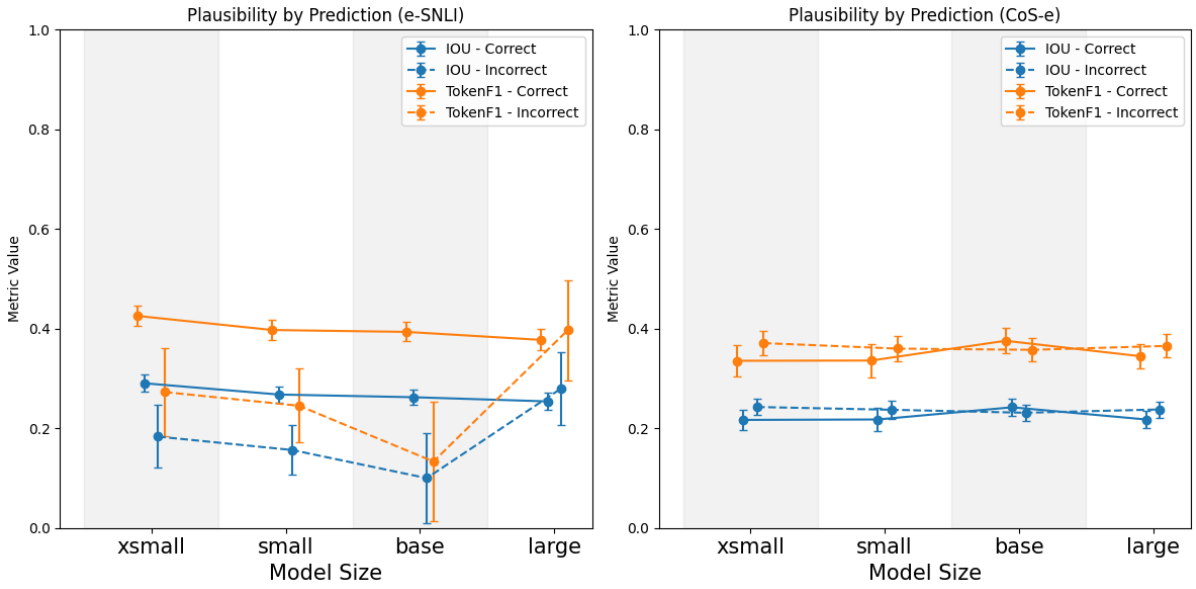


Figure 4.10: Plausibility by prediction for e-SNLI (left) and CoS-e (right).

4.3 Chapter Summary

With the first experiment, we illustrated how performance improved with increasing model size. We explained local instances from e-SNLI and CoS-e to build an intuition on the faithfulness and plausibility metrics. Finally, we explained 100 instances from each dataset and reported on trends with respect to model size and with respect to labels and predictions. All results including mean standard errors are additionally captured in the form of tables and can be found in appendix A.

Our findings can be summarised as shown in the table below. In the following Discussion chapter, each finding will be critically analysed including limitations, hypotheses and future direction.

Finding	**Description**
	Comprehensiveness
1	Positively correlates with model size for CoS-e.
2	Positively correlates with model size when predictions are wrong.
3	Tends to be higher for contradictive pairs than for neutral pairs.
4	Increases with model size for contradiction pairs in MNLI and stagnates in e-SNLI.
	Sufficiency
5	Is uncorrelated with model size.
6	Tends to be lower for contradictive pairs than for entailment pairs.
	IOU and TokenF1
7	Are uncorrelated with model size.
8	Are significantly lower for neutral sentence pairs than for entailment and contradiction pairs.
9	Are higher for correct predictions than for incorrect predictions in e-SNLI.
10	Are independent of the correctness of the prediction in CoS-e.

Table 4.9: Summary of findings.

Chapter 5

Discussion

In this discussion chapter we critically analyse the findings from the experiments and identify limitations of our approach. Based on the limitaitons we propose future work that could further contribute to this research.

5.1 Findings Analysis

In this section, each of the 10 findings is critically discussed. We provide hypotheses that can be tested in future work and outline limitations and implications associated our findings.

5.1.1 Comprehensiveness

- **Finding 1: Comprehensiveness positively correlates with model size for CoS-e.**

  In the zero shot setting, we find that comprehensiveness improves monotonically with model size from 0.304 for the smallest to 0.391 for the largest model. Comprehensiveness increases across both correct and incorrect predictions. One potential hypothesis is that comprehensiveness increases with the general performance of the model. We have seen a similar monotonic increase in performance for CoS-e which could be related to the increase in comprehensiveness. The reason why we don't see the same trend for NLI could be related to an unbalanced distribution between

correct and incorrect instances. Future work could further test this hypothesis by including other zero shot classification datasets with an increased number of explanations.

- **Finding 2: Comprehensiveness positively correlates with model size when predictions are wrong**

  Even though comprehensiveness slightly increases with model size for both NLI datasets we in fact found that comprehensiveness only increased for incorrect predictions. One hypothesis could be that a larger model provides inherently more comprehensive explanations for wrong predictions than for correct predictions. Another hypothesis is that larger models tend to be more confident in their predictions when they are wrong which could result in bigger change in predictions when explanations are removed from the input. The fact that correct and incorrect predictions were strongly imbalanced due to high performance, however, limits this finding. Potential future work could investigate explanations that contain the same number of correct predictions as incorrect predictions to rule out confounding effects.

- **Finding 3: Comprehensiveness tends to be higher for contradictive pairs than for neutral pairs**

  For both NLI datasets, we observe that comprehensiveness tends to be higher for contradiction instances than for neutral instances. The intuition behind that phenomenon is that if the explanation contains contradictive words then removing the explanation is likely to result in a change in prediction to either neutral or entailment. However, removing words from a neutral sentence does not necessarily change the prediction. The prediction can still be neutral. This illustrates a central limitation of the comprehensiveness measure in an NLI setting. We suggest that future work should not apply comprehensiveness to explain neutral sentence pairs in NLI. More generally, this implies that the applicability of comprehensiveness to

measure explainability is task-dependent.

- **Finding 4: Comprehensiveness increases with model size for contradiction pairs in MNLI and stagnates in e-SNLI**

  This finding seems ambiguous as both datasets cover the same task. The difference might be due to dissimilarities between MNLI and e-SNLI. MNLI is considered more diverse as it covers multiple topics in both written and spoken language. SNLI on the other hand is based on image captions only. Testing other NLI datasets can help to identify a more consistent trend. Note that for e-SNLI the error bars in the contradictive pairs are overlapping which limits the reliability of the stagnating behaviour. If the number of explained instances increases the metric would become more robust.

5.1.2 Sufficiency

- **Finding 5: Sufficiency is uncorrelated with model size**

  Across all datasets, we cannot observe any trend regarding the interplay between sufficiency and model size. Even after splitting the predictions by label sufficiency neither increases nor decreases systematically when model size varies. We do find an increase of sufficiency for incorrect predictions in the NLI setting however due to the low sample size for incorrect predictions any conclusions are unreliable which is undermined by the overlapping error bars. We hypothesise that sufficiency for LIME explanations is unaffected by model size. Potentially the same holds for any post hoc explainability technique which could be verified by repeating this experiment on different explainability techniques. Again, we suggest balancing the number of correct and incorrect predictions to draw more reliable conclusions.

- **Finding 6: Sufficiency tends to be lower for contradictive pairs than for entailment pairs**

Similar to comprehensiveness the best sufficiency on the NLI datasets is achieved on the contradictive sentence pairs while the LIME explanations on entailment pairs tend to have higher sufficiency. We hypothesise that it generally takes fewer words to explain a contradictive sentence pair than an entailment pair which could explain the difference. Note that the limitation of comprehensiveness regarding undesirable behaviour for neutral sentence pairs does not transfer to sufficiency. Consider highlights for a neutral instance. Predicting on highlights solely may result in low sufficiency in case the highlights reflect the neutral nature of the sentence pair well. Another hypothesis that could be tested in future work is that sufficiency is less task-dependent than comprehensiveness which would imply that sufficiency generally captures faithfulness better than comprehensiveness.

5.1.3 IOU and TokenF1

TokenF1 did not provide any additional information compared to IOU as both metrics align parallel to each other in every plot which is why we can reason about both metrics at once.

- **Finding 7: IOU and TokenF1 are uncorrelated with model size**

  All conducted experiments have neither shown a significant increase nor a decrease in the plausibility metrics when model size varies. A potential hypothesis that explains this observation would be that the plausibility of LIME explanations is independent of model size. Interestingly this would imply that the agreement with human-annotated highlight does not improve with model performance suggesting an inherent misalignment between the generated explanations and the true internal decision process. Future work could test if the plausibility of other explainability techniques are similarly independent of model size and performance. Including other human-annotated datasets would further help to verify our hypothesis.

- **Finding 8: IOU and TokenF1 are significantly lower for neutral sentence pairs than for entailment and contradiction pairs**

  For e-SNLI, we found that the plausibility metrics result in a lower score on neutral pairs than on entailment and contradiction pairs. We claim that fewer highlighted words are required to uniquely explain a contradiction or entailment instance compared to a neutral instance which makes an overlap with the human annotation more likely. Since both premise and hypothesis are unrelated in neutral pairs the correct highlighted tokens to justify that relationship are less objective and may vary between different annotators. Collecting multiple explanations by independent human annotators could verify that claim. We conclude that measuring plausibility depends on the classification task and its labels.

- **Finding 9: IOU and TokenF1 are higher for correct predictions than for incorrect predictions in e-SNLI**

  For e-SNLI, we observe that except from the large model the agreement between the LIME explanations and the human annotations is significantly better for correct predictions than for incorrect predictions which appears to be trivial since the LIME explanations are always taken with respect to the predicted class. We claim that the dip in the base model and peak in the large model for wrong prediction is due to the low sample size. As shown in table A.5 out of 100 explained instances only 5 were falsely predicted by the base model and 7 by the large model. We hypothesise that balancing out the prediction correctness would result in a similar plausibility gap between correct and incorrect predictions as for the smaller models.

- **Finding 10: IOU and TokenF1 are independent of the correctness of the prediction in CoS-e**

  We would expect a similar result as the previous finding for the CoS-e dataset. However, we observe that the plausibility metrics seem to be independent of the

prediction correctness. Given that 503 out of 1221 validation instances contained highlights covering the complete input sequence, however, this gives rise to the general quality of the human-annotated highlights in CoS-e. The paper that proposed this dataset only ensured that annotators "cannot move forward if they do not highlight any relevant words in the question" [14]. A more in-depth quality check of the human annotations may give more insights into the reliability of human annotations.

5.2 General Limitations and Future Work

This section discusses more general limitations of our approach and provides future directions that can help to address these limitations.

Our experiments only use one explainability technique (LIME) and only one transformer architecture (DeBERTaV3) of four different sizes. Involving other highlight-based explainability techniques (e.g. Anchors, SHAP, Integrated Gradients) and other transformer models (e.g. RoBERTa, GPT2, T5) can help to get further insights into transformer explainability. Our results suggest that the applicability of the faithfulness metrics depends on the task and its labels. Involving other datasets and tasks (e.g. named entity recognition, sentiment analysis) can verify this observation. To extend our experiments for plausibility other datasets with human-annotated highlights can be found here [5]. Note that in this project we calculated LIME explanations with respect to the predicted label. It might be worth repeating the experiments with respect to the true label instead. It would also be beneficial to include random explanations similar to [5] to better contextualise and interpret the metrics. Lastly, all results become more reliable when the number of explained instances increases. Our experiment used 100 LIME explanations for each dataset and each model. To mitigate confounding effects when comparing within labels or predictions one could select a balanced number of instances to explain across

labels and predictions instead of random selection. Note that, for perturbation-based techniques, computational limits will always be a potential bottleneck.

Regarding computational intensity, we observe LIME becomes increasingly computationally expensive when the model size grows as depicted in table 5.1. The reason for that is that for every explained instance a potentially large number of perturbed samples have to be predicted.

	MNLI	e-SNLI	CoS-e
xsmall	2min 3s	1min 8s	34min 35s
small	2min 40s	1min 40s	44min 28s
base	5min 20s	3min 35s	1h 27min 7s
large	15min 38s	12min 45s	4h 35min 50s

Table 5.1: Compute time for 100 LIME explanations on Nvidia's T4 GPU, 51GB RAM.

Especially, the zero shot classification setting requires significantly more computational resources with over 4 hours to calculate 100 LIME explanations on the large model. The reasons for longer compute times for zero shot classification are likely related to Huggingface's internal translation into an NLI setting where each candidate label requires its own forward pass. While explanations for larger models tend to be more comprehensive it might become impractical to compute them with LIME when models become too large. The same holds for other perturbation-based explainability techniques like SHAP or Anchors.

Our approach in this study was to evaluate explanations quantitatively to be able to reason about the quality of explanations more objectively. However, the findings suggest that token-level explanations might not be able to comprehensively explain an instance and higher-level concepts might be required that are not necessarily part of the input sequence. Another drawback of using highlight-based explanations is that they are not particularly human-centred and can be easily misinterpreted. A study found that out of 197 data scientists only a few "were able to accurately describe the visualizations

output by these [highlight based] tools" and that "data scientists over-trust and misuse interpretability tools" [95] which gives rise to more human understandable approaches like natural language explanations [10]. The caveat of natural language explanations is their subjectivity and their lack of quantitative evaluations. Overall, one could state that for NLP explainability there is a general tradeoff between objectiveness and human interpretability.

Measuring explainability objectively has no standard or established framework. Our project used comprehensiveness and sufficiency to evaluate faithfulness and IOU and TokenF1 to evaluate plausibility. Other approaches worth exploring include:

1. Comparing the explanation on a black box model with an inherently explainable model as a ground truth [96].

2. Conducting an agreement or concordance test between multiple explainability techniques where a model is considered explainable if the explanation of multiple methods aligns with each other [97]

Finally, despite their popularity posthoc explainability generally comes with limitations. Apart from the computational costs and overtrust, the explanations are always approximations that might not reflect the true decision boundaries. A study by Rudin [98] argues posthoc explanations "must be wrong" because if they were perfectly faithful to the black box model then the black box model would not be needed in the first place, only the explanation. As a result, posthoc explanations might be misleading and oversimplifying. Rudin therefore advocates "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead".

Chapter 6

Conclusion

This final chapter summarizes the presented work and provides some ethical considerations that put AI explainability into a broader context.

6.1 Summary

This project first provided an in-depth review of the literature related to transformer-based language models and AI explainability. We used pre-trained and fine-tuned DeBERTaV3 models of four different sizes and found improved performance for the larger models. The project involved three datasets (MNLI, e-SNLI, CoS-e) covering two NLP tasks, namely natural language inference and zero shot classification. LIME was used as a highlight-based posthoc explainability method. The explanations were evaluated based on their faithfulness and plausibility. We measured faithfulness by the comprehensiveness and sufficiency metrics and measured plausibility by intersection over union and token level f1 scores. To build an intuition for those metrics we evaluated two local instances before scaling up the number of explanations to 100 for each dataset. On a global level, we found that comprehensiveness positively correlates with model size in the zero shot classification setting suggesting improving faithfulness for larger models. Sufficiency seemed to be uncorrelated with model size across all datasets. In the NLI setting, we found evidence that both comprehensiveness and sufficiency strongly vary based on the label and prediction correctness. We, therefore, suggest to assess the applicability of both metrics depending on the task and its labels. Both plausibility metrics

were uncorrelated with model size. Given that performance increased with model size this raises questions on why agreement with human annotations does not increase. We suggest that there is some extent of misalignment between the model's internal decision process and its posthoc explanation.

In sum, this project contributes by (1) applying two approaches to quantitatively evaluate explainability while there is no common framework, by (2) novelly investigating the interplay between explainability and model size in state-of-the-art transformer models and by (3) providing an extensible code repository with outlined future directions.

6.2 Ethical Considerations

This last section aims to embed our work into a broader context and provides ethical considerations that are important for future development in the field of transformer explainability.

Language models are already surpassing human performance on several tasks [42] and tend to further improve as developers increase the number of parameters, training data size and the amount of computing power for training [6]. When language models replace human decision-making in high-stakes applications like criminal justice or healthcare, explainability becomes crucial.

Our experiments on plausibility indicate that while larger models exhibit improved performance, their agreement with human explanations regarding why a prediction was made does not improve as the model size increases. This phenomenon has previously been observed in a study called "Language Models Don't Always Say What They Think" [99]. The study found that the prediction in chain of thought prompting can be manipulated although the explanations sound plausible which poses a risk of overtrusting large language models.

This phenomenon is closely related to the alignment problem [24] and the hallucination problem [23] which we already touched upon in the introduction. If we fail to faithfully explain the decisions of language models we risk that AI systems pursue unintended and unexpected objectives rather than agreeing with human values which is at the heart of the alignment problem. Furthermore, explainability could play a central role in understanding why language models hallucinate, that is understanding why language models often provide responses that do not align with reality. As of today hallucination and alignment are unsolved challenges in AI and require future efforts.

As discussed in the literature review the current trend is clearly to further scale up language models as increased scale translates to increased performance [6]. One obvious implication is the immense environmental footprint that comes with the hardware energy consumption for training these large models. A study from 2019 estimates that "training BERT on GPU is roughly equivalent to a trans-American flight" [100]. BERT has 340 million parameters. Imagining the carbon footprint for training an allegedly 100 trillion [21] parameter GPT-4 model raises general concerns from an environmental perspective. Note that our experiments have also shown that explaining large language models with perturbation-based techniques such as LIME also comes with high computational costs. The study [100] further emphasizes how financial requirements in terms of hardware infrastructure and cloud computing to produce paper-worth results drive a privatisation of AI research which makes it harder for academic institutions to pursue large-scale computing projects.

In conclusion, explainability in language models will play a crucial role regarding safety issues in language models. Furthermore, the scaling trend in language models relies on increasing energy consumption and financial requirements which calls for a more sustainable development in NLP research from a social, economic and environmental perspective.

Appendix A

Additional Tables

Dataset	Model Size	Metric			
		Accuracy	Precision	Recall	F1
MNLI	xsmall	0.87835 (0.87162, 0.88467)	0.87785 (0.87158, 0.88435)	0.87822 (0.8714, 0.88458)	0.87776 (0.87121, 0.8837)
	small	0.87794 (0.87121, 0.88406)	0.87697 (0.87084, 0.88369)	0.87743 (0.87083, 0.88393)	0.87708 (0.87057, 0.88317)
	base	0.89995 (0.89404, 0.90565)	0.90003 (0.89436, 0.90588)	0.9002 (0.89457, 0.90574)	0.89971 (0.89378, 0.90556)
	large	0.90199 (0.89628, 0.90769)	0.90183 (0.89578, 0.90765)	0.90219 (0.89601, 0.908)	0.90169 (0.89601, 0.90737)
e-SNLI	xsmall	0.92034 (0.91485, 0.92562)	0.92025 (0.91455, 0.92543)	0.92016 (0.91495, 0.92509)	0.9202 (0.91478, 0.92533)
	small	0.92197 (0.91688, 0.92735)	0.92192 (0.91664, 0.92708)	0.92178 (0.91602, 0.92698)	0.92184 (0.9162, 0.92732)
	base	0.93142 (0.92654, 0.9366)	0.93141 (0.92635, 0.93628)	0.93131 (0.92627, 0.93607)	0.93134 (0.92656, 0.93618)
	large	0.93213 (0.92725, 0.93721)	0.93214 (0.92738, 0.93695)	0.93204 (0.92665, 0.93691)	0.93207 (0.92733, 0.93675)
CoS-e	xsmall	0.33088 (0.30467, 0.3579)	– –	– –	– –
	small	0.33579 (0.30876, 0.362)	– –	– –	– –
	base	0.35872 (0.33251, 0.38329)	– –	– –	– –
	large	0.37756 (0.35053, 0.40541)	– –	– –	– –

Table A.1: Model performance evaluation on validation sets with 95% confidence intervals.

Dataset	Model Size	Label	Faithfulness		Plausibility	
			↑ Comprehensiveness	↓ Sufficiency	↑ IOU	↑ TokenF1
MNLI	xsmall	contradiction	0.810 (+- 0.038)	0.071 (+- 0.035)	-	-
		entailment	0.759 (+- 0.042)	0.257 (+- 0.034)	-	-
		neutral	0.788 (+- 0.034)	0.139 (+- 0.048)	-	-
	small	contradiction	0.853 (+- 0.041)	0.072 (+- 0.030)	-	-
		entailment	0.805 (+- 0.039)	0.213 (+- 0.038)	-	-
		neutral	0.794 (+- 0.035)	0.098 (+- 0.036)	-	-
	base	contradiction	0.895 (+- 0.034)	0.120 (+- 0.039)	-	-
		entailment	0.768 (+- 0.050)	0.284 (+- 0.045)	-	-
		neutral	0.728 (+- 0.049)	0.201 (+- 0.042)	-	-
	large	contradiction	0.939 (+- 0.018)	0.154 (+- 0.039)	-	-
		entailment	0.750 (+- 0.057)	0.295 (+- 0.042)	-	-
		neutral	0.789 (+- 0.046)	0.107 (+- 0.036)	-	-
e-SNLI	xsmall	contradiction	0.805 (+- 0.034)	0.078 (+- 0.026)	0.289 (+- 0.031)	0.418 (+- 0.039)
		entailment	0.713 (+- 0.039)	0.234 (+- 0.030)	0.315 (+- 0.025)	0.463 (+- 0.028)
		neutral	0.663 (+- 0.035)	0.129 (+- 0.030)	0.244 (+- 0.028)	0.363 (+- 0.037)
	small	contradiction	0.744 (+- 0.042)	0.130 (+- 0.032)	0.286 (+- 0.029)	0.422 (+- 0.033)
		entailment	0.783 (+- 0.051)	0.227 (+- 0.039)	0.249 (+- 0.025)	0.378 (+- 0.033)
		neutral	0.652 (+- 0.040)	0.245 (+- 0.049)	0.242 (+- 0.030)	0.357 (+- 0.038)
	base	contradiction	0.808 (+- 0.038)	0.064 (+- 0.020)	0.264 (+- 0.027)	0.395 (+- 0.033)
		entailment	0.786 (+- 0.043)	0.361 (+- 0.047)	0.291 (+- 0.025)	0.433 (+- 0.030)
		neutral	0.701 (+- 0.045)	0.143 (+- 0.025)	0.211 (+- 0.028)	0.319 (+- 0.037)
	large	contradiction	0.759 (+- 0.046)	0.092 (+- 0.034)	0.259 (+- 0.024)	0.391 (+- 0.032)
		entailment	0.809 (+- 0.038)	0.302 (+- 0.038)	0.292 (+- 0.026)	0.434 (+- 0.029)
		neutral	0.768 (+- 0.042)	0.197 (+- 0.040)	0.220 (+- 0.036)	0.317 (+- 0.044)

Table A.2: Macro faithfulness and plausibility metrics by label with mean standard errors on 100 explained instances.

Dataset	Model Size	Prediction	Faithfulness		Plausibility	
			↑ Comprehensiveness	↓ Sufficiency	↑ IOU	↑ TokenF1
MNLI	xsmall	Correct	0.792 (+- 0.025)	0.171 (+- 0.024)	-	-
		Incorrect	0.743 (+- 0.051)	0.098 (+- 0.080)	-	-
	small	Correct	0.813 (+- 0.025)	0.143 (+- 0.022)	-	-
		Incorrect	0.841 (+- 0.052)	0.065 (+- 0.059)	-	-
	base	Correct	0.790 (+- 0.030)	0.192 (+- 0.025)	-	-
		Incorrect	0.849 (+- 0.045)	0.325 (+- 0.107)	-	-
	large	Correct	0.813 (+- 0.030)	0.184 (+- 0.024)	-	-
		Incorrect	0.925 (+- 0.032)	0.245 (+- 0.103)	-	-
e-SNLI	xsmall	Correct	0.732 (+- 0.023)	0.151 (+- 0.018)	0.290 (+- 0.017)	0.425 (+- 0.021)
		Incorrect	0.654 (+- 0.061)	0.085 (+- 0.078)	0.183 (+- 0.063)	0.273 (+- 0.088)
	small	Correct	0.722 (+- 0.028)	0.186 (+- 0.024)	0.268 (+- 0.017)	0.397 (+- 0.021)
		Incorrect	0.756 (+- 0.060)	0.379 (+- 0.100)	0.156 (+- 0.049)	0.245 (+- 0.074)
	base	Correct	0.768 (+- 0.026)	0.178 (+- 0.023)	0.262 (+- 0.016)	0.393 (+- 0.019)
		Incorrect	0.691 (+- 0.095)	0.365 (+- 0.081)	0.100 (+- 0.089)	0.133 (+- 0.119)
	large	Correct	0.772 (+- 0.026)	0.183 (+- 0.023)	0.254 (+- 0.018)	0.377 (+- 0.022)
		Incorrect	0.852 (+- 0.034)	0.368 (+- 0.116)	0.280 (+- 0.074)	0.397 (+- 0.100)
CoS-e	xsmall	Correct	0.281 (+- 0.027)	-0.111 (+- 0.021)	0.217 (+- 0.021)	0.336 (+- 0.031)
		Incorrect	0.319 (+- 0.024)	-0.104 (+- 0.019)	0.243 (+- 0.017)	0.371 (+- 0.024)
	small	Correct	0.319 (+- 0.033)	-0.181 (+- 0.033)	0.218 (+- 0.023)	0.336 (+- 0.034)
		Incorrect	0.314 (+- 0.023)	-0.124 (+- 0.026)	0.237 (+- 0.018)	0.360 (+- 0.025)
	base	Correct	0.369 (+- 0.029)	-0.092 (+- 0.026)	0.242 (+- 0.018)	0.376 (+- 0.025)
		Incorrect	0.348 (+- 0.027)	-0.038 (+- 0.025)	0.231 (+- 0.016)	0.358 (+- 0.023)
	large	Correct	0.396 (+- 0.026)	-0.101 (+- 0.035)	0.218 (+- 0.017)	0.345 (+- 0.025)
		Incorrect	0.388 (+- 0.031)	-0.068 (+- 0.027)	0.238 (+- 0.016)	0.365 (+- 0.023)

Table A.3: Macro faithfulness and plausibility metrics by prediction with mean standard errors on 100 explained instances.

Dataset	Contradiction	Entailment	Neutral
MNLI	32	36	32
e-SNLI	33	32	35

Table A.4: Number of observations by label for MNLI and e-SNLI for 100 explained instances.

Dataset	Model Size	Correct	Incorrect
MNLI	xsmall	85	15
	small	85	15
	base	90	10
	large	91	9
e-SNLI	xsmall	92	8
	small	92	8
	base	95	5
	large	93	7
CoS-e	xsmall	39	61
	small	34	66
	base	39	61
	large	37	63

Table A.5: Number of correct and incorrect predictions per dataset for 100 explained instances.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[2] "Huggingface nlp course." Accessed: 2023-08-10 at https://huggingface.co/learn/nlp-course/chapter1/4?fw=pt.

[3] C. Molnar, "Interpretable machine learning: A guide for making black box models explainable," 2022. Accessed: 2023-08-13 at https://christophm.github.io/interpretable-ml-book.

[4] M. T. Ribeiro, S. Singh, and C. Guestrin, """ why should i trust you?" explaining the predictions of any classifier," in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1135–1144, 2016.

[5] J. DeYoung, S. Jain, N. F. Rajani, E. Lehman, C. Xiong, R. Socher, and B. C. Wallace, "ERASER: A benchmark to evaluate rationalized NLP models," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, (Online), pp. 4443–4458, Association for Computational Linguistics, July 2020.

[6] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," arXiv preprint arXiv:2001.08361, 2020.

[7] J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn, and P. Villalobos, "Compute trends across three eras of machine learning," in 2022 International Joint Conference on Neural Networks (IJCNN), pp. 1–8, IEEE, 2022.

[8] "Large language models: A new moore's law?." Accessed: 2023-08-18 at https://huggingface.co/blog/large-language-models.

[9] G. Ciatto, M. I. Schumacher, A. Omicini, and D. Calvaresi, "Agent-based explanations in ai: Towards an abstract framework," in International workshop on explainable, transparent autonomous agents and multi-agent systems, pp. 3–20, Springer, 2020.

[10] O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom, "e-snli: Natural language inference with natural language explanations," Advances in Neural Information Processing Systems, vol. 31, 2018.

[11] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), (New Orleans, Louisiana), pp. 1112–1122, Association for Computational Linguistics, June 2018.

[12] P. He, J. Gao, and W. Chen, "Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing," arXiv preprint arXiv:2111.09543, 2021.

[13] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, (Lisbon, Portugal), pp. 632–642, Association for Computational Linguistics, Sept. 2015.

[14] S. Aggarwal, D. Mandowara, V. Agrawal, D. Khandelwal, P. Singla, and D. Garg, "Explanations for CommonsenseQA: New Dataset and Models," in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the

11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), (Online), pp. 3050–3065, Association for Computational Linguistics, Aug. 2021.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

[16] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al., "Highly accurate protein structure prediction with alphafold," Nature, vol. 596, no. 7873, pp. 583–589, 2021.

[17] C. B. Anfinsen, "Principles that govern the folding of protein chains," Science, vol. 181, no. 4096, pp. 223–230, 1973.

[18] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27, pp. 270–279, Springer, 2018.

[19] "Glue benchmark leaderboard." Accessed: 2023-08-17 at https://gluebenchmark.com/leaderboard.

[20] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., "Language models are few-shot learners," Advances in neural information processing systems, vol. 33, pp. 1877–1901, 2020.

[21] "A new chip cluster will make massive ai models possible." Accessed: 2023-08-18 at https://www.wired.com/story/cerebras-chip-cluster-neural-networks-ai/.

[22] A. Madsen, S. Reddy, and S. Chandar, "Post-hoc interpretability for neural nlp: A survey," ACM Computing Surveys, vol. 55, no. 8, pp. 1–42, 2022.

[23] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," ACM Computing Surveys, vol. 55, no. 12, pp. 1–38, 2023.

[24] R. Ngo, L. Chan, and S. Mindermann, "The alignment problem from a deep learning perspective," arXiv preprint arXiv:2209.00626, 2022.

[25] W. Samek and K.-R. Müller, "Towards explainable artificial intelligence," Explainable AI: interpreting, explaining and visualizing deep learning, pp. 5–22, 2019.

[26] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," arXiv preprint arXiv:1606.06565, 2016.

[27] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, (Red Hook, NY, USA), pp. 3323–3331, Curran Associates Inc., 2016.

[28] F. Doshi-Velez, M. Kortz, R. Budish, C. Bavitz, S. Gershman, D. O'Brien, K. Scott, S. Schieber, J. Waldo, D. Weinberger, et al., "Accountability of ai under the law: The role of explanation," arXiv preprint arXiv:1711.01134, 2017.

[29] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden debt in machine learning systems," Advances in neural information processing systems, vol. 28, 2015.

[30] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," A Practical Guide, 1st Ed., Cham: Springer International Publishing, vol. 10, no. 3152676, pp. 10–5555, 2017.

[31] "Machine bias there's software used across the country to predict future criminals. and it's biased against blacks.." Accessed: 2023-08-18 at https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[32] J. Dressel and H. Farid, "The accuracy, fairness, and limits of predicting recidivism," Science advances, vol. 4, no. 1, p. eaao5580, 2018.

[33] C. S. Chan, H. Kong, and L. Guanqing, "A comparative study of faithfulness metrics for model interpretability methods," in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), (Dublin, Ireland), pp. 5029–5038, Association for Computational Linguistics, May 2022.

[34] "Huggingface debertav3 models." Accessed: 2023-08-18 at
https://huggingface.co/cross-encoder/nli-deberta-v3-xsmall,
https://huggingface.co/cross-encoder/nli-deberta-v3-small,
https://huggingface.co/cross-encoder/nli-deberta-v3-base,
https://huggingface.co/cross-encoder/nli-deberta-v3-large.

[35] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," AI Open, vol. 3, pp. 111–132, 2022.

[36] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 2, pp. 604–624, 2021.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[38] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE transactions on neural networks, vol. 5, no. 2, pp. 157–166, 1994.

[39] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," arXiv preprint arXiv:1607.06450, 2016.

[40] J. Alammar, "The illustrated transformer," 2018. Accessed: 2023-08-04 at https://jalammar.github.io/illustrated-transformer.

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[42] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," Science China Technological Sciences, vol. 63, no. 10, pp. 1872–1897, 2020.

[43] R. Liu, Y. Shi, C. Ji, and M. Jia, "A survey of sentiment analysis based on transfer learning," IEEE access, vol. 7, pp. 85401–85412, 2019.

[44] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, (Seattle, Washington, USA), pp. 1631–1642, Association for Computational Linguistics, Oct. 2013.

[45] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Advances in Neural Information Processing Systems (C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, eds.), vol. 26, Curran Associates, Inc., 2013.

[46] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, Oct. 2014.

[47] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, (Brussels, Belgium), pp. 353–355, Association for Computational Linguistics, Nov. 2018.

[48] X. Song, A. Salcianu, Y. Song, D. Dopson, and D. Zhou, "Fast WordPiece tokenization," in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, (Online and Punta Cana, Dominican Republic), pp. 2089–2103, Association for Computational Linguistics, Nov. 2021.

[49] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in Proceedings of the IEEE international conference on computer vision, pp. 19–27, 2015.

[50] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in International Conference on Learning Representations, 2020.

[51] L. Zhuang, L. Wayne, S. Ya, and Z. Jun, "A robustly optimized BERT pre-training approach with post-training," in Proceedings of the 20th Chinese National Conference on Computational Linguistics, (Huhhot, China), pp. 1218–1227, Chinese Information Processing Society of China, Aug. 2021.

[52] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: Decoding-enhanced bert with disen-

tangled attention," in 2021 International Conference on Learning Representations, May 2021. Under review.

[53] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, 2019.

[54] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., "Improving language understanding by generative pre-training," OpenAI, 2018.

[55] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., "Language models are unsupervised multitask learners," OpenAI, 2019.

[56] "Openai chatgpt." Accessed: 2023-08-17 at https://chat.openai.com/.

[57] OpenAI, "Gpt-4 technical report," arXiv preprint arXiv:2303.08774, 2023.

[58] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, (Online), pp. 7871–7880, Association for Computational Linguistics, 2020.

[59] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," The Journal of Machine Learning Research, vol. 21, no. 1, pp. 5485–5551, 2020.

[60] O. Biran and C. Cotton, "Explanation and justification in machine learning: A survey," in IJCAI-17 workshop on explainable AI (XAI), vol. 8, pp. 8–13, 2017.

[61] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in

2018 IEEE 5th International Conference on data science and advanced analytics (DSAA), pp. 80–89, IEEE, 2018.

[62] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in Proceedings of the AAAI conference on artificial intelligence, vol. 32, 2018.

[63] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, (Red Hook, NY, USA), pp. 4768–4777, Curran Associates Inc., 2017.

[64] L. S. Shapley, A Value for N-Person Games. Santa Monica, CA: RAND Corporation, 1952.

[65] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAT* '20, (New York, NY, USA), pp. 607–617, Association for Computing Machinery, 2020.

[66] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in Workshop at International Conference on Learning Representations, 2014.

[67] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, pp. 3319–3328, JMLR.org, 2017.

[68] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," arXiv preprint arXiv:1706.03825, 2017.

[69] S. Sanyal and X. Ren, "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), pp. 10285–10299, Association for Computational Linguistics, Nov. 2021.

[70] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," The Annals of Statistics, vol. 29, no. 5, pp. 1189 – 1232, 2001.

[71] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously.," J. Mach. Learn. Res., vol. 20, no. 177, pp. 1–81, 2019.

[72] S. Wiegreffe and A. Marasovic, "Teach me to explain: A review of datasets for explainable natural language processing," in Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (J. Vanschoren and S. Yeung, eds.), vol. 1, Curran, 2021.

[73] S. Vashishth, S. Upadhyay, G. S. Tomar, and M. Faruqui, "Attention interpretability across nlp tasks," ArXiv, vol. abs/1909.11218, 2019.

[74] S. Jain and B. C. Wallace, "Attention is not Explanation," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), (Minneapolis, Minnesota), pp. 3543–3556, Association for Computational Linguistics, June 2019.

[75] S. Serrano and N. A. Smith, "Is attention interpretable?," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, (Florence, Italy), pp. 2931–2951, Association for Computational Linguistics, July 2019.

[76] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. van Keulen, and C. Seifert, "From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai," ACM Computing Surveys, vol. 55, no. 13s, pp. 1–42, 2023.

[77] R. Andrews, J. Diederich, and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks," Knowledge-based systems, vol. 8, no. 6, pp. 373–389, 1995.

[78] W. Silva, K. Fernandes, M. J. Cardoso, and J. S. Cardoso, "Towards complementary explanations using deep neural networks," in Understanding and Interpreting Machine Learning in Medical Image Computing Applications: First International Workshops, MLCN 2018, DLF 2018, and iMIMIC 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16-20, 2018, Proceedings 1, pp. 133–140, Springer, 2018.

[79] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein, "A diagnostic study of explainability techniques for text classification," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), (Online), pp. 3256–3274, Association for Computational Linguistics, Nov. 2020.

[80] J. Strout, Y. Zhang, and R. Mooney, "Do human rationales improve machine explanations?," in Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, (Florence, Italy), pp. 56–62, Association for Computational Linguistics, Aug. 2019.

[81] N. F. Rajani, B. McCann, C. Xiong, and R. Socher, "Explain yourself! leveraging language models for commonsense reasoning," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, (Florence, Italy), pp. 4932–4942, Association for Computational Linguistics, July 2019.

[82] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), (Berlin, Germany), pp. 1715–1725, Association for Computational Linguistics, Aug. 2016.

[83] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," in ICLR, 2020.

[84] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for SQuAD," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), (Melbourne, Australia), pp. 784–789, Association for Computational Linguistics, July 2018.

[85] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 11 2019.

[86] W. Yin, J. Hay, and D. Roth, "Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), (Hong Kong, China), pp. 3914–3923, Association for Computational Linguistics, Nov. 2019.

[87] T. W. Bhavitvya Malik, Patrick von Platen, "Multinli huggingface," 2018. Accessed: 2023-07-02 at https://huggingface.co/datasets/multi_nli.

[88] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in Proceedings of the IEEE international conference on computer vision, pp. 2641–2649, 2015.

[89] D. O.-M. Camburu, "Cos-e huggingface," 2018. Accessed: 2023-07-15 at https://github.com/OanaMariaCamburu/e-SNLI.

[90] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "CommonsenseQA: A question answering challenge targeting commonsense knowledge," 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), pp. 4149–4158, Association for Computational Linguistics, June 2019.

[91] T. W. Lewis Tunstall, Patrick von Platen and Q. Lhoest, "Cos-e huggingface," 2019. Accessed: 2023-07-08 at https://huggingface.co/datasets/cos_e.

[92] J. Wang, J. Tuyls, E. Wallace, and S. Singh, "Gradient-based analysis of NLP models is manipulable," in Findings of the Association for Computational Linguistics: EMNLP 2020, (Online), pp. 247–258, Association for Computational Linguistics, Nov. 2020.

[93] M. T. C. Ribeiro, "Lime github," 2016. Accessed: 2023-07-11 at https://github.com/marcotcr/lime.

[94] D. B. Rosen, "How to calculate confidence intervals for performance metrics in machine learning using an automatic bootstrap method," 2021. Accessed: 2023-09-16 at https://towardsdatascience.com/get-confidence-intervals-for-any-model-performance-metrics-in-machine-learning-f9e72a3becb2.

[95] H. Kaur, H. Nori, S. Jenkins, R. Caruana, H. Wallach, and J. Wortman Vaughan, "Interpreting interpretability: understanding data scientists' use of interpretability tools for machine learning," in Proceedings of the 2020 CHI conference on human factors in computing systems, pp. 1–14, 2020.

[96] M. Naylor, C. French, S. Terker, and U. Kamath, "Quantifying explainability in nlp and analyzing algorithms for performance-explainability tradeoff," arXiv preprint arXiv:2107.05693, 2021.

[97] D. Alvarez-Melis and T. S. Jaakkola, "A causal framework for explaining the predictions of black-box sequence-to-sequence models," arXiv preprint arXiv:1707.01943, 2017.

[98] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," Nature machine intelligence, vol. 1, no. 5, pp. 206–215, 2019.

[99] M. Turpin, J. Michael, E. Perez, and S. R. Bowman, "Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting," arXiv preprint arXiv:2305.04388, 2023.

[100] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," arXiv preprint arXiv:1906.02243, 2019.