ST. PÖLTEN UNIVERSITY
OF APPLIED SCIENCES

Informatik & Security
fh st.pölten

Comparing Explainability Concepts for Transformer-based Language Models at the example of Counter Speech and Hate Speech Classification

Diplomarbeit

zur Erlangung des akademischen Grades

Diplom-Ingenieur/in

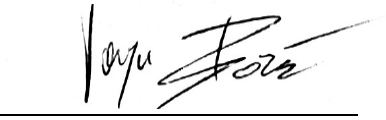
eingereicht von

Jaqueline Böck BSc

im Rahmen des
Studiengangs Data Intelligence an der Fachhochschule St. Pölten

Betreuung
Betreuer/Betreuerin:
FH-Prof. Priv.-Doz. Dipl.-Ing. Mag. Dr. Matthias Zeppelzauer
Dipl.-Ing. Armin Kirchknopf, BA MA BSc

St. Pölten, 12.09.2023



(Unterschrift Autor*in)

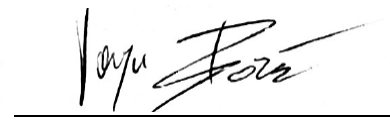
1. Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Diplomarbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich sonst keiner unerlaubten Hilfe bedient habe.
- ich dieses Diplomarbeitsthema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.
- diese Arbeit mit der vom/von der Begutachter*in beurteilten Arbeit übereinstimmt.

Der Studierende/Absolvent*in räumt der FH St. Pölten das Recht ein, die Diplomarbeit für Lehre- und Forschungstätigkeiten zu verwenden und damit zu werben (z.B. bei der Projektevernissage, in Publikationen, auf der Homepage), wobei der/die Absolvent*in als Urheber zu nennen ist. Jegliche kommerzielle Verwertung/Nutzung bedarf einer weiteren Vereinbarung zwischen dem/der Studierenden/Absolvent*in und der FH St. Pölten.

St. Pölten, 12.09.2023



(Unterschrift Autor*in)

2. Acknowledgement

First and foremost, I would like to express my deepest gratitude to my advisors, FH-Prof. Dipl.-Ing. Mag. Dr. Matthias Zeppelzauer and Dipl.-Ing. Armin Kirchknopf, BA MA BSc, of the Institute of Creative\Media/Technologies at the University of Applied Sciences in St. Pölten. They have consistently supported me throughout the creation of this work. No matter the hour, they were always available and willing to listen, offering both academic guidance and personal encouragement. In addition, I would like to thank the entire ICMT team, and my fellow students who, during moments of self-doubt and stress, consistently found ways, whether through conversations or leisure activities, to redirect my focus and morale.

I would also like to extend my heartfelt thanks to my sister, Annika, my parents, and my best friend, Resi. They have been my pillar of strength, providing motivation and support even during the most challenging times.

3. Abstract

The rise of digitalization has brought with it numerous issues, including online threats, insults, incitement, and hate speech [1]. The use of counter speech by internet users is becoming increasingly important to combat the mentioned issues. As a result, the monitoring of social media and its online activity is necessary for a save internet use and earns great importance in today's world [2]. Several machine learning models have been used for text classification and detection tasks in the past. Especially newer Transformer models like BERT [6] show state-of-the-art results in identifying hateful content on social media. These networks are capable to handle large amounts of data by achieving high accuracy scores in the evaluation procedure [1]. However, due to their complex structure, these models often lack explainability and interpretability. Even though explainability has become one of the most present topics in Artificial Intelligence over the past few years [2]. The research in counter speech detection combined with explainability approaches are limited. Based on this, after a thorough research of suitable datasets and the training of BERT Transformer models for the detection of hate- and counter speech, different explainability methods to explain the decisions of the model are compared [5]. The evaluation focuses on three distinct explainability methods: the well-established LIME [9] and Integrated Gradients [10], and two emerging techniques, GlobEnc [11] and ProtoTEx [12].

Three BERT Transformer models have been fine-tuned on two counter speech related tasks. Also, one BERT model for language detection task was used to evaluate the ProtoTEx model. However, early evaluation of this method showed that the ProtoTEx model was not suitable for this comparative study and was discharged from further evaluation. The other three methods are evaluated regarding their faithfulness, plausibility, understandability, sufficiency, trustworthiness, satisfaction, and help-/usefulness. Even though, all the remaining three methods can be considered as plausible, none of them fulfilled the other criteria to a sufficient extend. Only the LIME methods showes some tendencies in satisfying the understandability and the sufficiency criteria.

Kurzfassung

Die zunehmende Digitalisierung hat zahlreiche Probleme mit sich gebracht wie online Übergriffen, welche oft mit Drohungen, Beleidigungen und Hassreden einher gehen [1]. Der Einsatz von Gegenrede durch Internetnutzer*innen wird immer wichtiger, um den genannten Problemen entgegenzuwirken. Folglich ist die Überwachung sozialer Medien und ihrer Online-Aktivitäten für eine sichere Internetnutzung notwendig und wird immer wichtiger [2]. In der Vergangenheit wurden verschiedene Modelle des maschinellen Lernens für Textklassifizierungs- und Erkennungsaufgaben verwendet. Insbesondere neuere Transformer-Modelle wie BERT [6] zeigen besonders hervorstechende Ergebnisse bei der Erkennung von hasserfüllten Inhalten in sozialen Medien. Diese Netzwerke sind in der Lage, große Datenmengen zu verarbeiten und erreichen hohe Performanz im Evaluierungsverfahren [1]. Aufgrund ihrer komplexen Struktur mangelt es diesen Modellen jedoch häufig an Erklärbarkeit und Interpretierbarkeit. Dabei ist die Erklärbarkeit in den letzten Jahren zu einem der aktuellen Themen in der Künstlichen Intelligenz geworden [2]. Die Forschung auf dem Gebiet der Erkennung von Gegenredner*innen in Kombination mit Erklärungsansätzen ist jedoch begrenzt. Folglich wurden nach einer gründlichen Recherche zu geeigneten Datensätze und dem Trainieren von BERT-Transformer-Modellen zur Erkennung von Hass- und Gegenrede verschiedene Erklärbarkeitsmethoden miteinander verglichen, um die Entscheidungen der Modelle besser verständlich und für den Menschen interpretierbar zu machen [5]. Die Evaluierung konzentriert sich auf drei verschiedene Erklärungsmethoden: die bewährten LIME [9] und Integrated Gradients [10] method sowie zwei neue Techniken, GlobEnc [11] und ProtoTEx [12]. Drei BERT-Transformer-Modelle wurden für zwei Aufgaben bezüglich der Erkennung von Gegenrede feinabgestimmt. Eines der BERT-Modelle wurde dafür trainiert, um zwischen Deutscher und Englischer Sprache in Texten zu unterscheiden, um mit diesem Modell die ProtoTEx-Methode zu bewerten. Die frühe Bewertung dieser Methode zeigte jedoch, dass das trainierte ProtoTEx-Modelle für diese vergleichende Studie nicht geeignet sind, und wurden aus der weiteren Bewertung ausgeschlossen. Die anderen drei Methoden wurden hinsichtlich mehrerer Kriterien wie Treue, Plausibilität, Verständlichkeit, Suffizienz, Vertrauenswürdigkeit, Zufriedenheit und Nützlichkeit mit Hilfe einer Ablationsstudie und einer Benutzer*innenstudie bewertet. Obwohl alle drei verbleibenden Methoden als plausibel angesehen werden können, erfüllte keine von ihnen die verbleibenden Kriterien im ausreichenden Maße. Lediglich die LIME-Methode zeigte gewisse Tendenzen, die Kriterien der Verständlichkeit und der Suffizienz zu erfüllen.

4. Table of Contents

5. Introduction

5.1. Motivation

Threats, insults, incitement, and hate-filled postings - they are among the biggest problems that comes with digitalization [1]. Countering hate is one strategy for combating hate speech. Therefore, counter speakers on social media play a significant role in the dynamic and safety of social media platforms like Twitter, Instagram and TikTok. As these platforms become increasingly prominent in today's society, monitoring them is growing in importance [3]. However, manually monitoring such websites is time-consuming, requires significant human resources, and exposes monitors to potentially traumatizing content. Therefore, additionally to countering such hateful content, it is of special interest to develop Machine Learning (ML) methods which support filtering such controversial posts in the first place when it comes to detecting hate speech itself. Additionally, to hate speech monitoring, detecting counter speakers can be beneficial for promoting them to continue and even produce more counter speech [1]. This can even lead to the dynamic that they motivate and recruit new counter speakers to combat against the hate on social media. Also, finding persons which already do counter hateful postings, but in an inappropriate way (e.g.: using insults or hate) can be found and provided with examples for more ethically correct and better counter responses [4].

Recent automatic monitoring approaches only focus on detecting hateful content in social media. However, this is a quite difficult task for ML models since the information that can be obtained by such online mediums is often short, noisy, unstructured and lack of proper manner [5]. Traditional methods often lack in capturing the complex and diverse features present in social media data. Therefore, such traditional methods may not fully capture the significance or structure of words and sentences in comments, which makes them often unsuitable for the complex task of hate speech detection and in the following, the detection of counter speech. However, research has demonstrated that more advanced models using attention mechanisms can selectively focus on important and unimportant parts of text [6]. For text classification tasks, such attention-based models currently provide the best results. One of the most popular models are Bidirectional Encoder Representations from Transformers (BERT) [7], which became increasingly popular for classification and detection tasks in social media due to their ability to effectively capture the context and semantic meaning of texts. These models can handle large amounts of data and have achieved state-of-the-art results in various Natural Language Processing (NLP) tasks. BERT is more adaptable for the use case since it does not need as much computational power as other Transformer models, which it is particularly interesting for the task of detecting counter speech in social media posts and is therefore the model architecture chosen within this thesis.

However, one of the major problems of such complex Transformer models and their decisions is that they are often not explainable and/or traceable. This means that their inner

workings and decisions are not reasonable and understandable for humans which is extremely important in real life applications and for building trust in the algorithm [2]. Depending on the ethical significance of the task, it is essential to know on what basis the algorithm makes its decision on in order to verify its trustworthiness and for making sure that their decisions are fair and unbiased [8]. Several papers have been used for comparing explainability methods for models using textual data. However, none of them focuses of the explainability for models with the focus on counter speech detection.
Due to these circumstances, the scope of this thesis is to fine-tune BERT Transformer models for detecting counter speech and to investigate in different explainability methods to make its decisions more explainable to humans. For this comparative study, four explainability methods, LIME [9] Integrated Gradients [10], GlobEnc [11] and ProtoTEx [12] have been chosen and evaluated by an ablation study and a user study.

5.2. Research Questions

The main research questions to be answered in this thesis are:

- *Which explainability methods can be used to explain the decisions of Transformer-based language models?*
  The field of XAI is a broad one ranging from traditional methods to more advanced deep learning models. Especially Transformer models consist of highly complex architectures and their decisions are often not interpretable for humans. Therefore, investigation in well working explainability techniques for Transformer models is needed.
  To answer this question, a comprehensive literature review on existing XAI methodologies suitable for Transformer architectures is pursued. The aim is to determine four distinct methods, each based on a unique strategy. These selected methods will be evaluated on BERT Transformer models that will be fine-tuned for the purpose of detecting counter speech in text.

- *Which explainability methods are most helpful in result interpretation for humans?*
  The need of transparency in the decision-making process of ML architectures is crucial especially for sensitive tasks. Therefore, ensuring that humans can understand the behavior of a ML model is paramount to build trust in the model. This question focuses on examining which explainability concepts are most beneficial for human users. The chosen explainability methods will be assessed by evaluating them through both an ablation study and a user study. The evaluation is based on criteria predefined in the context of this research, focusing on making interpreting the results of the classifier more interpretable for humans.

- *How do different explainability concepts differ and what are the strengths and weaknesses?*
  As the field of XAI grows, several concepts have emerged to make models in NLP (especially Transformer models) more explainable. The variety of methods make it necessary to evaluate their strengths and weaknesses to not only gather knowledge of the restrictions and benefits, but also for developing further and more advances techniques in the future. According to a previously done literature review and the comparative study, the advantages and disadvantaged of the chosen explainability concepts and the four chosen methods are analyzed.

5.3. Methodological Approach

For answering the stated research questions mentioned in the previous section, an in-depth literature research and a comparison of four explainability methods which can be adapted to BERT-like Transformers is pursued. The chosen methods include one attention-based (GlobEnc [11]) one gradient-based (Integrated Gradients [10]), one perturbation-based (LIME [9]) and one approach based on prototypes (ProtoTEx [12]).

At the beginning, a literature review on existing approaches on counter speech detection and related datasets will be conducted. A state-of-the-art analysis of different XAI methods for Transformer models will be done. A thorough literature research on Google and databases like IEEE, ScienceDirect, SpringerLink, GoogleScholar and ACM was performed. For essential coding resources, datasets, and models, huggingface.co and Github.com are considered. Even though no particular language restriction is set, the focus is on English and German resources. The main keywords/phrases and wildcards that are used within the search strategy include:
BERT, Transformer, explainable AI (XAI), explainab*, interpretab*, Natural Language Processing, NLP, counter speech detection, hate speech detection, gradient based, attention based, attribution methods, prototypes, text classification/detection. ChatGPT was used during the writing process of the master thesis to paraphrase some quotations.

Even though there are several studies which conduct comparisons of XAI methods [13]–[17] (for other tasks then counter speech detection), only two of the found papers include a prototype-based approach in their experiments for Transformer explainability [12], [18]. Also, previous papers claim that recent, traditional techniques come with certain difficulties and limitation which makes it important that newer and more variations of approaches get developed and evaluated. Some of these limitations are mentioned and evaluated in the course of this thesis.
Four methods have been chosen of four different methodological concepts, an attention-based method, a gradient-based method, a perturbation-based method, and a method that utilizes the use of prototypes are selected for the comparative study. The method based on gradients is Integrated Gradients [19] and the perturbation-based approach is LIME [9]. The attention-based method which is used is GlobEnc [11], which is different than recent attention-based approaches, since it integrated all the elements that are part of the encoder block and accumulates them throughout all layers. For the prototype-based approach,

representative examples for each provided class in the dataset are generated in order to provide the user a better understanding in why a certain prediction is made by the model [12]. For this approach, the implementation of ProtoTEx [12] is adapted to the BERT model for counter speech detection.

A comprehensive assessment of methods to evaluate the advantages and disadvantages of explaining BERT Transformers to humans is carried out. This evaluation will hinge on several criteria regarding to the explainability of ML models. To verify the criteria of plausibility [20], a ablation study is employed. To evaluate the criteria of faithfulness [20], understandability and trustworthiness [21], [22], sufficiency [23], satisfaction [24] and usefulness/helpfulness [24] a user study is performed. This study comprises two primary tasks: a forward prediction task and a comparative analysis. The participants include persons with a certain knowledge of Artificial Intelligence (AI) and explainable AI.

Since the use case of this thesis is the detection of counter speech in text, a literature review on appropriate datasets is pursued. The chosen counter speech related datasets are the Thou Shalt Not Hate dataset [25] and the HateCounter dataset [26], where the Thou Shalt Not Hate dataset includes the classes “counter” and “non-counter” and the HateCounter one consists of the classes “hate speech” and counter speech”. To account for weaknesses in the evaluation of ProtoTex, a third dataset, the Europarl dataset [13] is included. This dataset is considered for the straightforward task of distinguishing between the German and English languages. A well-balanced subsample of this data is created. Each of the datasets is used to fine-tune a BERT model for the respective task.

The fine-tuned Transformer models are evaluated using several metrics, including accuracy, F1-Score, recall, and precision [24]. For the ablation study, we will calculate and visualize the Pearson correlation and the changes in prediction performance. The user study will be assessed based on standard deviation, effect size (Cohen's d), and confidence intervals. Furthermore, an additional ANOVA analysis is planned to provide deeper insights into the results.

5.4. Structure of the Thesis

After stating the problem and the approach of this paper, the following will describe the structure of the thesis in more detail. Chapter 6 offers a theoretical overview of ML models in NLP, transitioning from traditional to deep learning models. Within this, the spotlight is on the introduction and various concepts of attention mechanisms. After discussing the forward pass and backpropagation functions in ML, an introduction to BERT Transformer models and their architectures is presented. Subsequently, pivotal terminologies and criteria in explainable AI (XAI) are addressed, encompassing topics like global and local explanations, post-hoc, ante-hoc, and self-explanation. The chapter concludes with a comprehensive literature review on different explainability approaches, particularly those employed for textual data and Transformer models.

The following chapter 6.3.3 focuses on the concepts of XAI methods that can be applied on Transformer models and its decisions. The four concepts selected in this thesis are thoroughly described. Each of these concepts aligns with a method tailored for the classification models discussed in this paper. These four methodologies, alongside their functionalities, are further elaborated in section 6.3.3. After the theoretical background the experimental setup is described in section 8. At first, the use case of counter speech detection and its significance is elaborated. This leads to a detailed discourse on selected datasets, training strategies for BERT-based classifiers, and the technical implementation—including hardware, software, and package selections.
The evaluation procedure is described in detail and employs several metrics for comparing the classification results, a user study and an ablation study for accessing the results of the XAI methods. The results of this evaluation are found in chapter 9, accompanied by a thorough discussion on the results and insights garnered during implementation and evaluation. The last chapter 10 gives an overview of the discoveries, addressing the research questions and weighting pros and cons of the various methodologies of the different approaches. Finally, future work for upcoming future research topics is stated.

6. Background and State-of-the-Art

This section provides an introduction of the background of Natural Language Processing (NLP) in terms of text classification and explores the state-of-the-art of current explainability methods in NLP.
At the beginning, the background and developments in terms of text classification are discussed including traditional as well as deep learning approaches. Light is shed on the training functionality of ML models by providing information on the forward pass and the backpropagation. Also, an introduction on attention mechanisms and the variations of the concept of computing attention in a deep learning model are given. By the end of this chapter, the Transformer architecture and its concepts are explained.

6.1. Background on Text Classification

Text classification is an essential task in NLP. It includes tagging text with predefined classes or labels, which are determined by the model from the content. Numerous practical applications are based on text classification including sentiment analysis, spam detection or counter- and hate- speech detection on social media. In the past, traditional as well as deep learning-based methods have been introduced for the task of text classification as shown in Fig. 1.

In the initial stage, the textual data usually underwent preprocessing steps including word segmentation, data cleaning, and statistics. Traditional ML methods are in the need of suitable, often manually extracted features from the samples since these methods' efficiency is significantly constrained by the quality of the extracted features. Deep Learning models can extract the features by themselves and do not need any further intervention. More detail on the traditional and the Deep Learning Methods can be found in the following chapters 6.1.1 and 6.1.2. After training the chosen models on the classification task the models are tested on an unseen data split of their respective task. According to the predicted label and the true label of the dataset the model can be evaluated by computing several evaluation metrics e.g., accuracy or F1-score [27].

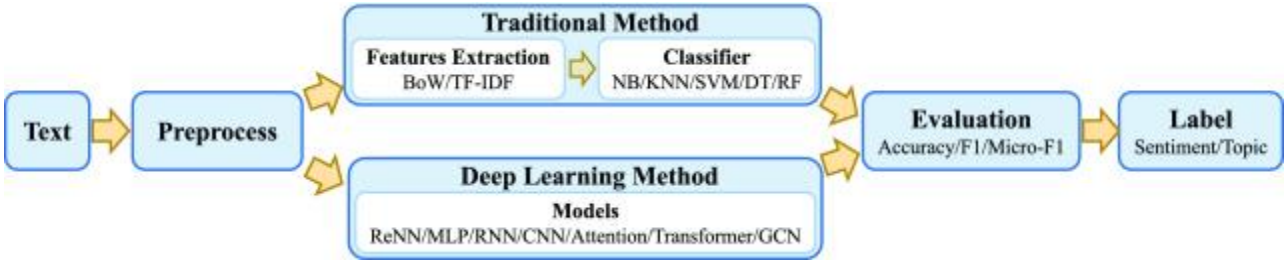


Fig. 1. Flowcharts of text classification approaches showing traditional methods with essential feature extraction and deep leaning methods. Adapted from [27].

6.1.1. Traditional Text Classification Methods

Traditional models typically rely on the manual extraction of meaningful features from the text samples which are afterwards processed by classical classification algorithms. In the early stages of text classification, traditional ML algorithms like Bag-of-Words (BoW) [28], N-gram [29] or TF-IDF (Term Frequency-Inverse Document Frequency) [30] were introduced to extract these features. The extracted features where then used as input to train a classification model.
In the early stages, rule-based methods were used which consist of conditional statements or “if-then” statements which define different relationships between the variables of the features. These algorithms are particularly useful, and they are also often interpretable by design [27], [31]. After this rule-based approaches, classical statistical algorithms like Naïve Bayes (NB) [32], Support Vector Machines (SVM) [33], K-Nearest Neighbours (KNNs) [34] and Decision Trees (DT) [33] were introduced. Compared to the rule-based approaches, these methods tend to have benefits regarding their stability and accuracy since their vocabularies might be incomplete due to the variation in different languages. But still, these methods rely on feature engineering which needs domain knowledge and tend to show issues in their scalability, which make these methods less feasible when large amounts of data are used. Also, traditional methods lack in understanding the semantic meaning of the given input sentences since they focus more on the syntactic representation of words. This behavior can cause them to miss out on understanding the context in which words are used, making them less effective for tasks that require a nuanced understanding of text [35]. Since 2010, people have started using deep learning models to classify text rather than using the traditional methods mentioned before [27].

6.1.2. Deep Learning Models

The use of Deep Neural Networks (DNNs) comes with the benefit that these methods can determine and learn semantic relationships in the data by their own without the need of human input. A variety of different input data can be analysed using these models including single label, multi label, unsupervised and unbalanced data.

With the rise of deep learning models, Neural Networks (NN) became popular in the field of NLP especially for the task of text classification. Convolution Neural Networks (CNNs) [36], Recurrent Neural Networks (RNNs) [37], including their more advanced variant, the Long-Short-Term Memory networks (LSTMs) had a rise withing the NLP community [38], [39]. RNNs are among the most popular networks for text classification since they can capture sequential dependencies of textual data and can retain information from earlier parts of the text during processing which leads to a certain understanding of the semantics of the text. However, a common issue that arises during the training of these RNNs is the vanishing gradient problem, which occurs especially in long sentences when the gradients of earlier parts become more insignificant for the RNN unit than later parts as illustrated in Fig. 2.

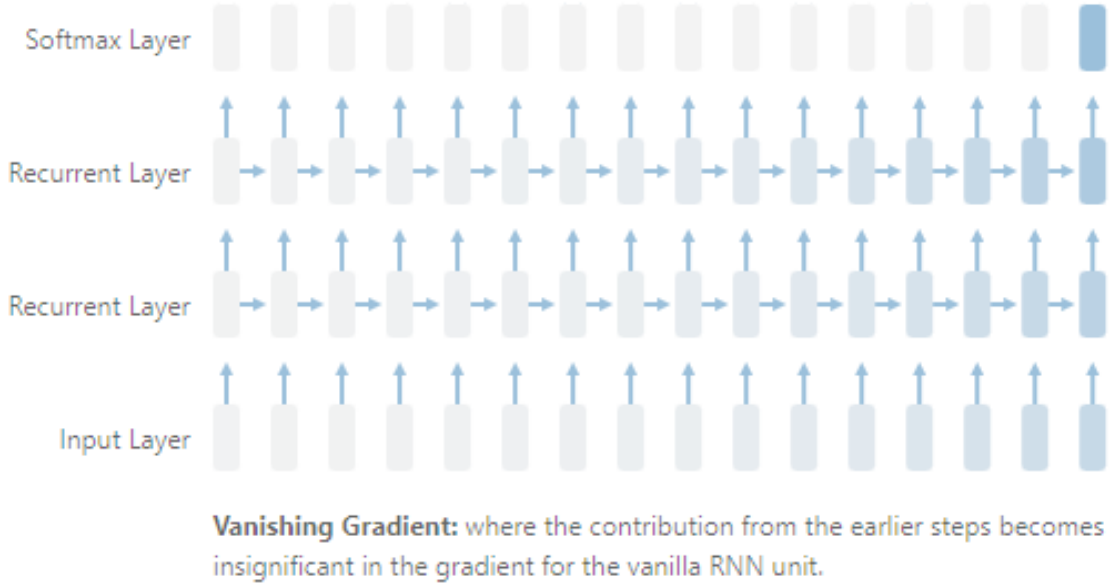


Fig. 2. Vanishing gradients in RNN units. Adapted from [40].

This issue usually occurs during the backpropagation process of the RNN architecture. During backpropagation, the weights of the model are adjusted by gradients which are computed by continuous multiplications of derivates.

Forward Pass

During the forward pass, the input data is presented as feature vector or a sequence of features in the input layer of the NN. This data flows through one or more hidden layers composed of neurons (nodes). These neurons are interconnected by weighted connections. The mathematical representation of this process in a single neuron can be given as:

$$z_j = \Sigma_i w_{ji} \times x_{ji} + b_j \quad (1)$$

Where:

- z_j represents the weighted sum for the neuron j
- w_j is the weight of neuron j
- x is the input data
- b_j is the bias of neuron j

Following this, each neuron applies an activation function to its weighted sum to produce its output. Activation functions introduce non-linearity into the network, enabling it to learn complex patterns and form non-linear mappings between inputs and outputs. Commonly used activation functions include $ReLU$ (Rectified Linear Units), $sigmoid$, $tanh$, and $softmax$ [41].

For a given weighted sum z_j , the output using an activation function (e.g., $sigmoid\ \sigma(z_j) =$) is:

$$\sigma(z_j) = \frac{1}{1 + e^{-z_j}} \tag{2}$$

The final outputs from these hidden layers are passed to the network's output layer, which produces the prediction of the network. Upon completion of the forward pass, the network's predicted output is compared to the target class, commonly referred to as the ground truth. This comparison uses a loss function C to quantify the difference between the predicted and actual values. One popular choice for this is the *Mean Squared Error (MSE)* [41], represented as:

$$C = MSE = \frac{1}{N}\Sigma_i(y_i - \hat{y}_i)^2 \tag{3}$$

Where:

- N is the number of data points
- y_i is the actual output
- $\hat{y}_i$ is the predicted label of the model

For simplicity we will use a general formulation of the loss function C, where y_i is the output label:

$$C = \frac{1}{2}(y_i - target)^2 \tag{4}$$

The factor $\frac{1}{2}$ is introduced to the squared error term so that when differentiated, the factor of 2 from the power rule cancels out the factor, leaving the gradient as just the difference between the predicted and actual value [42]. The primary goal during training is to minimize this loss [43], [44].

Backpropagation

Backpropagation, or the backward pass, is the method used to compute the gradients of the loss function C with respect to the parameters (weights and biases) of the network. The gradients provide both the magnitude and direction of changes required to minimize the loss—underlying the concept of gradient descent. During backpropagation, at first the gradient of the loss function is calculated with respect to the activations of the output layer where $\frac{\partial C}{\partial y_j}$ partial derivative of C with respect to the output y_j of the neuron j.

$$\frac{\partial C}{\partial y_j} = \left(y_j - target\right) \tag{5}$$

This information about the gradients is then propagated backward through all the layers of the network. The gradients of the loss function C concerning the weighted sums and outputs of the activation function for each neuron are computed. Using the chain rule, this gradient concerning the weight w_j is:

$$\frac{\partial C}{\partial w_j} = \frac{\partial C}{\partial y_j} \times \frac{\partial y_j}{\partial z_j} \times \frac{\partial z_j}{\partial w_{ji}} \tag{6}$$

Which can be simplified to:

$$\frac{\partial C}{\partial w_j} = \frac{\partial C}{\partial z_j} \times \frac{\partial z_j}{\partial w_{ji}} \tag{7}$$

Upon computing the gradients for the network parameters, optimizers, such as Stochastic Gradient Descent (SGD), use these gradients to update the parameters in the direction opposite to the gradient:

$$w_j = w_j - \eta \times x + \frac{\partial C}{\partial w_j} \tag{8}$$

Where η denotes the learning rate. During training, the forward pass and the backpropagation are iterative processed with the goal of gradually improving the network. The goal is to reach the point where the model converges. At this point the loss of the model minimized and the prediction of the model should improve and get more accurate [41].

One issue of RNNs is that they are limited in the information that can be captured of a text since they are only capable of accessing the past information because they process the sequences in a single direction. This is why they are also referred as unidirectional RNNs. A variant which tackles this issue are bidirectional RNNs. Bidirectional RNNs include two separate RNNs that process sequenced in both directions: one processed the sequence from the beginning to the end (forward), and the other one from the end to the beginning of the sequence (backward). These architectures can therefore capture contextual dependencies in texts and are particularly useful in tasks where understanding the future context is crucial [45]. Another issue of RNNs is the gradient vanishing problem which occurs when the calculated derivates are extremely small [46]. One contribution for tackling this vanishing gradient problem was the introduction of Long Short-Term Memory (LSTM). LSTMs are based on the idea of RNNs but include a cell which has the purpose of remembering values over arbitrary time intervals. They also consist of three gate structures to control the information flow of the network. Therefore, when compared to RNNs, LSTMs can capture longer context than RNNs [46].
Another important approach to improve RNNs was to add an encoder and a decoder to the model. These architectures were initially introduced for sequence-to-sequence tasks like text translation to improve capturing the long- and short- term dependencies between words. This is possible because these architectures can process input sequences sequentially [47]. However, a primary limitation with the basic encoder-decoder approach lies in the requirement for the encoder to compress all relevant information into a fixed-length vector. This compression becomes problematic, particularly for longer sentences, leading to a decline in performance as the input sentence length increases. Also, the needed high computational power of those models is an issue. To overcome these issues, researchers introduced the attention mechanism [48] which enables the model to dynamically focus of specific parts of the input sentence. Therefore, the need of compressing all relevant information into a fixed-length vector can be avoided. The attention mechanism used in the architecture of Transformer models has become one of the most relevant inventions in the past few years [48].

6.1.3. Attention Mechanism

Initially, the attention mechanism was introduced to enhance various architectures to allow them to focus on the most crucial parts of an input text. The attention mechanism was first introduced in Bahdanau et al. [48] for enhancing a RNN model with an added encoder and decoder and stacked RNN layers for the task of machine translation. The author proposed a solution to enable the decoder to focus on relevant input words of the sentence and to overcome the issue of fixed length encoding vectors. The attention mechanism addresses this problem by dynamically adjusting the focus on relevant parts of the input during decoding, enabling the decoder to access and use the information more efficiently, regardless of the input sequence's length or complexity. The calculation of the hidden states is shown in Eq. (9) where x_t is the source input and h_t is the hidden state at step t and f is a non-linear activation function.

$$h_t = f(x_t, h_{t-1}), t = 2, 3, \dots, n \tag{9}$$

For the decoder, the notion of the hidden states is s_t and the notion of the target output is y_t. The length of the sequence is denoted as t.

$$s_t = f(s_{t-1}, y_{t-1}, c_t), t = 1, \dots, m \tag{10}$$

The decoder includes a context vector c_i which calculates a weighted sum (using the alignment scores) of the encoders hidden states of the input sequence h_i (10).

$$c_t = \sum_{j=1}^{n} \alpha_{t,i} h_i \tag{11}$$

The alignment model calculates a score, $\alpha_{t,i}$ (weights) for every pair of input i and each output at position t (y_t, x_i), according on how well they match (12).

$$\alpha_{t,i} = align(y_t, x_i) = \frac{exp\ (score(s_{t-1}, h_i))}{\sum_{i\prime=1}^{n} exp\ (score(s_{t-1}, h_{i\prime}))} \tag{12}$$

The collection of weights $\{\alpha_{t,i}\}$, illustrate the significance of each source hidden state in relation to each output. In the study of Bahdanau et al. [48], the alignment score α is parametrized by a feedforward network containing a singular hidden layer. This network is simultaneously trained with other model sections. The scoring function utilises $tanh$ as non-linear activation function (13).

$$score(s_t, h_i) = v_\alpha^T tanh\ (W_\alpha [s_t; h_i]) \tag{13}$$

The final scoring function $score\ (s_t, h_i)$, describes the alignment score between the target hidden state (s_t) and the specific source hidden state h_i. The vector of weights v_α and the

weight matrix W_α which linearly combines the target hidden state s_t and the source hidden state h_i are learned during the training of the network [48], [49].

Attention mechanisms have become a popular subject in academic research, with numerous studies exploring its nuances and potential applications, distinguishing attention mechanism with different attributions. Common attributions that include different attention scoring functions are local/global and soft/hard and self-attention [50]–[52]. More information of the self-attention functionality can be found in the following chapter 6.1.4.

6.1.4. Self-Attention Networks and Scaled Dot-Product Attention

The attention mechanism in machine and deep learning enables models to concentrate selectively on particular segments of the input data during a task. It allocates different levels of focus to diverse portions of the input. Several types of attention mechanisms have been introduced in the past. The three main types are:

- **Self-attention:** this type of attention is often referred as intra-attention or internal attention. This mechanism operates within a single sequence by capturing the relationships and inter-dependencies among words and sub-words, allowing models to grasp the context and associations within data.
- **Encoder-decoder attention:** this attention mechanism operates between separate sequences; the encoder interprets the input while the decoder generates the output. Often referred to as inter-sequence attention, its core functionality is to seamlessly connect and transfer information from the input sequence to the output sequence.
- **Multi-head attention:** This approach uses several concurrent attention operations to look at different parts of the data at the same time, allowing the model to capture various aspects simultaneously [53].

This chapter focuses on the self-attention mechanism, particularly its variant, the scaled dot product attention, and the multi-head attention. An Introductions in the architectures, the mathematical concepts and processed in these attention mechanisms is given within this chapter.

Scaled Dot-Product Attention

This variant is a refinement of the traditional dot-product attention, which incorporates a scaling technique. This scaling aims to enhance training efficiency and model stability by ensuring optimal data utilization during the training process [53]. The most popular architecture that relies on self-attention is the Transformer architecture that has been introduced by Vaswani et al. [54]. In self-attention, the importance, and the meaning of a word in a sentence is calculated by relating it to different other words in the sentence.

In the architecture of Transformers, the mentioned above “scaled dot-product attention” is implemented for determining on which word the Transformer should focus on [53].

In Fig. 3, the illustration showcases the methodology behind computing the scaled dot-product attention. The subsequent sections will provide a comprehensive breakdown of each step involved.

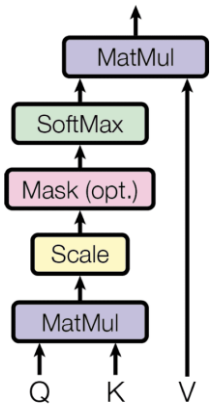
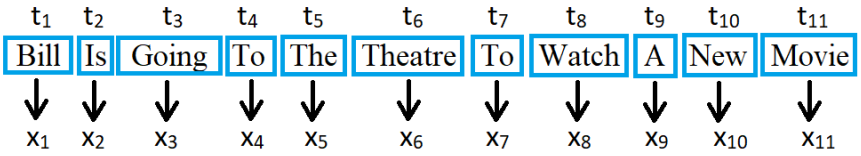


Fig. 3. Scaled Dot-Product Attention. Adapted from [43].

The process starts with the input sentence which is divided into individual word tokens. Tokens are individual pieces or units of a text, obtained by breaking down the text during the process of tokenization. In the context of NLP and text analysis, tokens are typically words, but they can also be phrases, sentences, or even individual characters, depending on the specific requirements of the task. Tokenization turns a series of characters into specific units that can later be represented numerically as vectors for NN analysis [55]. These vectors are called “embeddings”. In the embedding layer of the model, a transformation process unfolds. For every token in the input sequence t_i, individual embedding vectors x_i are generated as shown in Fig. 4 [43].



$$q_i = x_i W^q_{d\times d} \text{ for } i \in [1 \dots T]$$

$$k_i = x_i W^k_{d\times d} \text{ for } i \in [1 \dots T]$$

$$v_i = x_i W^v_{d\times d} \text{ for } i \in [1 \dots T]$$

Fig. 4. Illustration of transforming the tokens t_i to embedding vectors x_i, and calculating the query vectors q_i, key vectors k_i and value vectors v_i for each token of the sequence. Variable d donates the dimensionality of the embeddings and T is the sequence length. Adapted from [56].

The next step is to derive the query vectors q_i, the key vectors k_i and the value vectors V_i. The terms "queries", "keys", and "values" are originated from the context of database and information retrieval. In traditional databases, a "query" searches for a "key" to retrieve a "value". In attention mechanisms, the "query" vector identifies relevant "key" vectors from the input, and the corresponding "value" vectors provide the necessary information for the output [57].

Each of the embeddings x_i is multiplied with its weight matrix as shown in Fig. 4 to obtain the query vectors q_i, the key vectors k_i and the value vectors v_i.
In theory, the three distinct matrices, the query matrix Q, the key matrix K, and the value matrix V are calculated as followed:

- **Query Matrix (Q):** is obtained from the current focus point of the attention mechanism. The query vectors q_i are computed by multiplying the embedding vectors x_i with the value weight matrix W_q. After that, all query vectors q_i for every token are stacked together to get the value matrix of Q ($Q = [q_1; q_2; ...; q_i]$).
- **Key Matrix (K):** originates from each position within the input. It is computed by multiplying the embedding vector x_i with the learned value weight matrix W_k and stacking all key vectors k_i for every token together to get the value matrix of K ($K = [k_1; k_2; ...; k_i]$).
  Value Matrix (V): it holds information about each position in the input sequence. It is computed by multiplying the embeddings x_i with the learned value weight matrix W_v and stacking all query vectors v_i for every token together to get the value matrix of V ($V = [v_1; v_2; ... ; v_i]$) [43].

In practice, the embedding vectors of x_i are stacked together into a matrix of X, where every row of X corresponds to a token in the input sequence. This matrix X is then multiplied with the trained weight matrices W_k, W_k and W_v [58].

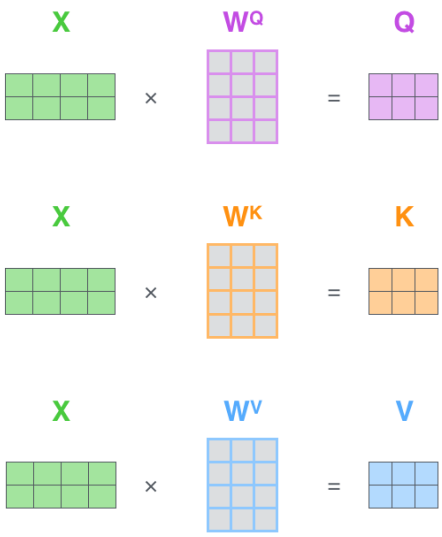


Fig. 5. Matrix multiplication of Q, K and V. Adapted from [52].

In the first matrix multiplication the similarities between the query matrix Q, the key matrix K and the value matrix V are calculated by the dot-product operation. To calculate a dot-product, the query vector's column size should match the key vector's row size. To achieve this the key vector gets transposed. After multiplying, we get the compatibility matrix as illustrated in Fig. 6.

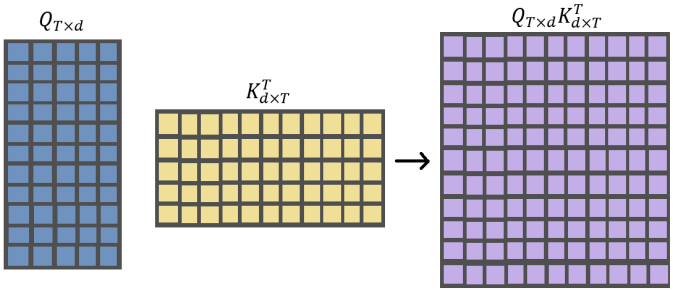


Fig. 6. First matrix multiplication of scaled dot-product attention. Adapted from [56].

The scaled dot-product of Vaswani et al. [54] is similar to the initial attention calculation of Luong et al. [51]. The only difference is that in the Transformer architecture the scaling factor is a factor of $\frac{1}{\sqrt{d_k}}$ to resolve the issue that the dot-products of QK can grow large in magnitude for large values of d_k. This would cause a vanishing gradient problem, since the $softmax$ would return extremely small gradients. Scaling the matrix helps to minimize the risk of this vanishing gradients. The author also claimed that the dot-product attention is more space-efficient and fast in practise due to its highly optimized matrix multiplication code.

A $softmax$ function is applied to normalize the scores over every key corresponding to a particular query, making sure the attention weights sum up to a total of 1. $Softmax$ plays a key role in the scaled dot-product attention by emphasizing which sections of the input are most relevant, assigning higher probabilities to those significant parts. The $softmax$ is applied to each of the rows of the scaled compatibility matrix. The resulting matrix is the attention matrix A. Each of the rows of this matrix will sum up to 1. In the final step, the attention matrix A (of size TxT) is multiplied with the value matrix V (size Txd) resulting in Eq. (14).

$$attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{14}$$

This yields the attention layer's output matrix (size Txd). The scaled dot-product attention's result, a combination of values shaped by how similar the queries are to the keys, is then passed to the next NN layer [56].

Multi-head Attention

Based on the scaled dot-product attention, the authors also introduced a multi-head attention architecture (Fig. 7) which projects a different learned projection of the queries q, keys k and values v for each head h. Instead of using a singular set of weight matrices W_Q, W_K, W_V to transform the input embedding X to Q, K, and V, the multi-head attention multiplies the weight matrices using h different sets. These are then concatenated and afterwards projected once more to produce the final output (the weighted representation of the input data) of the specific layer.

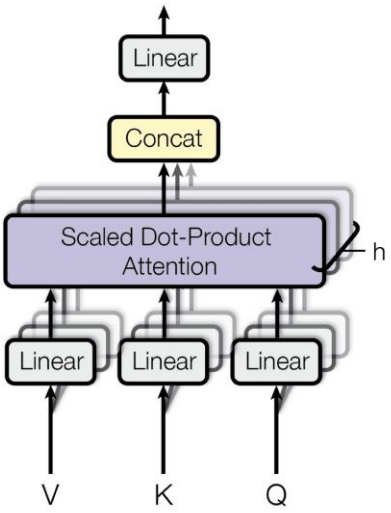


Fig. 7. Multi-Head Attention. Adapted from [54].

The multi-head attention function can be depicted as:

$$multihead(Q, K, V) = concat\ (head_1, \dots, head_h)W^O \tag{15}$$

Where the projection matrix of the multi-head output is W^O. Like all other weight matrices in the network, W^O is initialized (often with small random values) and then trained via backpropagation and gradient descent during the model's training process.
Each of the heads ($head_i, i, \dots, h$) uses a single attention function, each defined by its distinct learned projection matrices.

$$head_i = attention(QW_i^Q, KW_i^K, VW_i^V) \tag{16}$$

The Q_i, K_i, and V_i matrices are therefore calculated by multiplying the embeddings X_i with their specific weight matrices (W_i^Q, W_i^K, W_i^V) for each $head_i$. After the Q_i, K_i, and V_i matrices for each $head_i$, are calculated, they get scaled, and $softmax$ is added just as in the scaled dot-product attention mentioned above. All the resulting output heads Z_i are then concatenated ($head_i, i, \dots, h$).

The output of the layer is then generated by multiplying the combined outputs with the weight matrix W^O in a linear projection operation [59]. An illustration of the process is shown in Fig. 8.

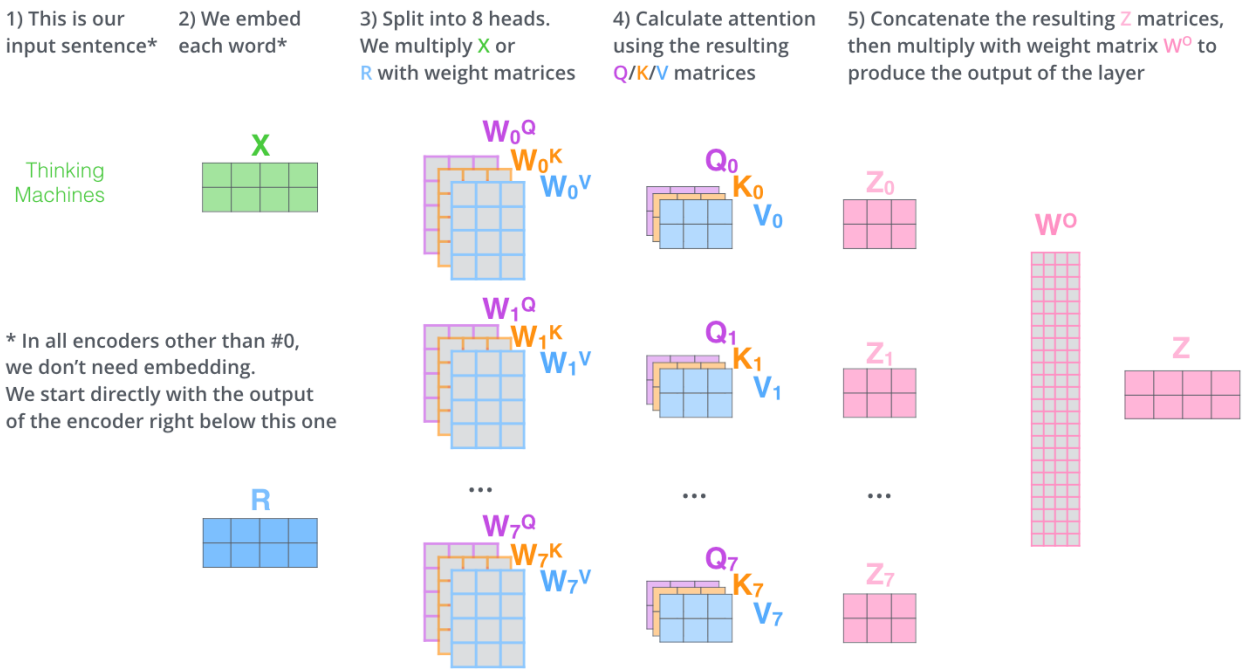


Fig. 8. Matrix multiplications of multi-head attention. Adapted from [52].

6.2. Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) is a language model first introduced in Devlin et al. in 2019 [7] which utilizes the Transformer architecture proposed by Vaswani et al. [54]. The initial Transformer architecture as shown in **Fig. 9** consists of an encoder and a decoder. In comparison, the BERT model only consists of the encoder part. In this section the Architecture of the initial Transformer architecture as well as the architecture of BERT are explained.

6.2.1. Initial Transformer Architecture

First, the training functionality of the general Transformer architecture will be explained on the example of translating a German text to English.

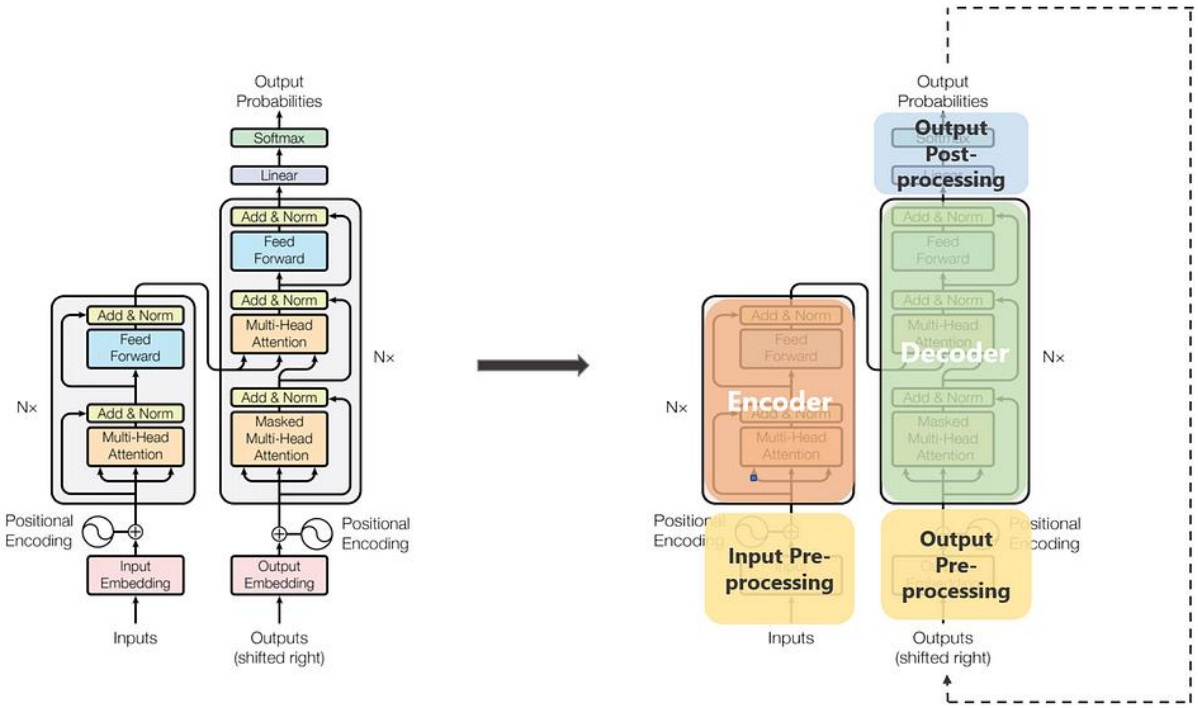


Fig. 9. Initial Transformer architecture by Vaswani et al. [54]. Adapted from [60].

Encoder

The preparation procedure for the input text is the same as for the scaled dot-product attention described in section 6.1.4. At first, the German sentence gets tokenized and fed into the encoder of the model. The encoder layer generates three embeddings for each word: the word embedding, positional embedding, and context embedding. The generated embeddings enter a multi-head attention layer. This layer generates attention vectors by

considering the relevance of all other words in the sentence. In creates a weighted contextual representation for each word. The attention mechanism (“multi-head attention”) operates using multiple heads to capture different types of relationships. Information about the functionality of this attention mechanism is also provided in section 6.1.4. Over each of the heads a weighted average of the attention vectors is computed to get the final attention vector which represent each word's comprehensive context in the sentence. The attention vectors are passed to a feedforward network, each token at a time. The feedforward network further transforms the representation vectors to make them more nuanced and capable of capturing complex relationships in the data. Each of the attention sets are independent from each other and can be parallelised. So, all words can be put at once in the encoder block and the result will be a set of vectors for every word.

Decoder

The same input preparation as in the encoder is done with the English sentence in the decoder. Here, the input sequence is shifted right, which means that each word position will only have access to previous words in the sequence, not the current or future words. The three obtained embeddings are put through a masked multi-head attention layer. While creating the English translation, this attention layer-block has access to all the words from the original German sentence, but it can only consider the English words that have been translated up to that point, not any words that come after. This limitation is important because it helps the model to actually learn to translate. Without it, the model might just immediately reveal the next word without really understanding the context. To achieve this, the "masking" technique is introduced, where future English words are hidden (or "masked") during the learning process. This is pursued using certain mathematical operations, which turn the values representing those future words to zero, making them invisible to the model at that step. This way, the model is encouraged to focus on the words it has seen so far, helping it to learn and generate better translations step by step. The obtained vectors, along with the attention vectors created by the encoder, are passed to an encoder-decoder attention block. The encoder-decoder attention block assesses the relationship between each English and German word vector and enables the translation process. At this point, each word captures its relationships and engagements with every other word in the respective sentences. Like in the encoder, the vectors are then passed through a feedforward network. The output layer is a linear layer (another feedforward layer) that expands the dimensions of the vectors to match the number of words in the English vocabulary, which then undergo a $softmax$ operation to form a probability distribution. The word with the highest probability is chosen as the output for that timestep. The decoder iterates through these steps, predicting one word at a time, until it generates an end-of-sentence token, indicating the completion of the translation [54], [61].

6.2.2. BERT-Transformer Architecture

The architecture of BERT consists of a stack of encoder layers where each of those comprises a self-attention mechanism, specifically a multi-head attention (section 6.1.4), and a linear feedforward network, just as the architecture in 6.2.1. The BERT architecture in shown in Fig. 10.

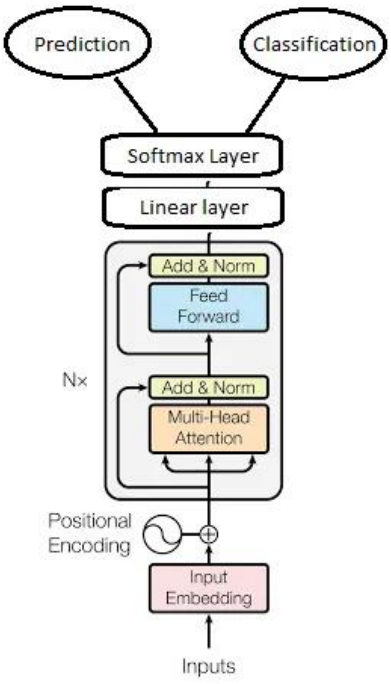


Fig. 10. BERT architecture by Devlin et al. in 2019 [7] with illustrated possible downstream tasks (word prediction or classification). Adapted from [62].

The multi-head attention block enables the BERT model to capture relationships between words in a text depending on the words surrounding. Also, for the BERT model the input text needs to be tokenized first before being put into the embedding layer. In tokenization, the words are split into sub-word tokens and additional special tokens are added. BERT uses the word-piece tokenizers concept for breaking down (unknown-) words into words the model might have in its vocabulary. For example, the word “reading” might be unknown to the model. When the word is tokenized to the sub-words “read” and “##ing”, “read” might be in the vocabulary of the model. “##” indicates that the sub-word is the part of another larger words. If the tokenizer is not able to break a unknown word down into sub-word tokens a special token [UNK] is added [63], [64]. The special tokens [CLS] and [SEP] are added to mark the beginning and the end of a sentence. The tokenized input text is transformed into token embeddings, segment encodings, and positional encodings within BERT's embedding layer as shown in Fig. 11.

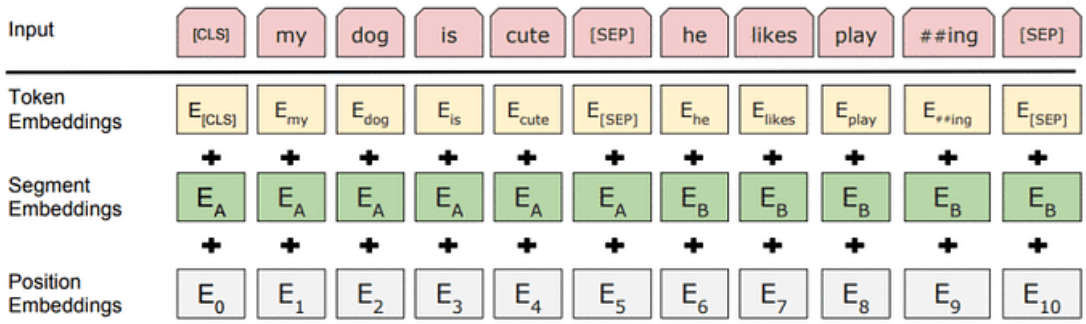


Fig. 11. BERT input representation. Adapted from [7].

These combined embeddings are then fed into the BERT model and processed in the encoder including its multi-head attention layer just as in the encoder block of the original Transformer architecture described in chapter 6.2.1.

Pre-training of BERT

Pre-training is a process where the model learns the underlying patterns, structures, and complexities in the data without being given explicit guidance or labelled examples. This is why it is also referred as “unsupervised”. Unsupervised pre-training enables BERT to enrich its language knowledge and to better generalize language representations capturing semantic and syntactic information of the data.

In the original paper of Devlin et al., [7] two pre-training strategies are proposed: Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). In Masked Language Modelling (MLM) (Fig. 12), the idea is that a specific percentage of the tokens from the input sentence are randomly replaced (“masked”) with [MASK] tokens.

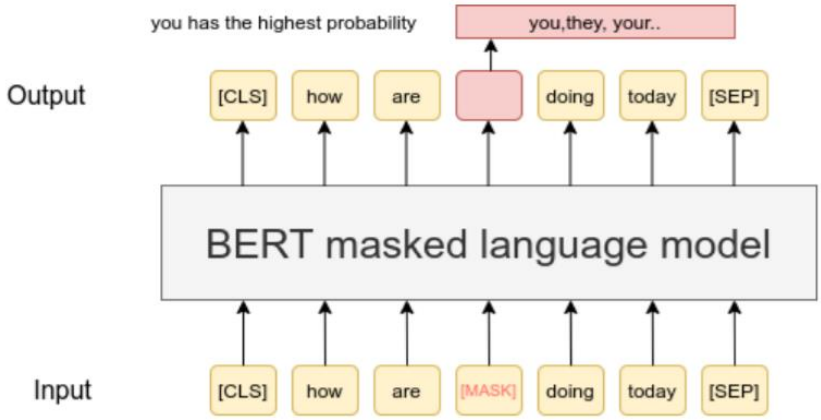


Fig. 12. BERT Training - Masked Language Modelling. Adapted from [65].

During training, BERT tries to predict the replaced token by considering all the other surrounding tokens of the text. In Next Sentence Prediction (NSP), pairs of sentences are presented to BERT (sentence A and sentence B) which are separated with [SEP] tokens. The segment encoding of the BERT encoder maintains the record of which token belongs to which sentence (A or B) and the position encoding stores the sentence numerical position for each token. BERT should predict whether the second one (sentence B) logically follows

the first sentence (sentence A) or not. However, in this training strategy, in some cases sentence B is not the actual next sentence of sentence A, but a randomly chosen sentence from a different part of the input sequence is. The idea is that the model should correctly determine if sentence B in indeed the next following sentence of A or not [7]. The initial BERT model is trained unlabelled texts from the BooksCorpus [66] that consists of 800M words and English Wikipedia texts consisting of 2,500M words. These texts provided the model with a strong foundation of language understanding.

Fine-tuning of BERT

To utilize BERT for a downstream task like text classification, an additional task-specific classification layer is added on top of the encoder stack. The generated word embeddings from the BERT encoder are then put through the classification layer and are mapped to the target classes. Depending on the task, the classification layer can differ in its architecture. A common practice is employing a fully connected layer followed by a *softmax* activation function for a multiclass problem. For a binary classification task, a *sigmoid* activation function is often used instead [67]. The main difference is that *sigmoid* is applied elementwise and produces an output of 0 to 1. The *softmax* function serves as normalization which takes a vector as input and outputs a vector of probabilities between 0 and 1 which always sum up to 1. During fine-tuning, a labelled dataset specific to the classification task is used to update the parameters (weights) of the models.

BERT is a highly complex architecture with many heavily interconnected individual components. Thus, one of the main challenges with BERT (as with other deep learning models) is that they are difficult to interpret, making it hard to understand the reasoning behind their predictions. To tackle this issue, various methods for XAI have been developed to clarify the decision-making process of Transformers [14], [68]. In the following sections, we will elaborate on the idea and concepts of explainable ML and how it can be applied to Transformer architectures.

6.3. Explainability in Machine Learning

The research of explainability in the field of ML is broad and the use of its definitions is not always clearly distinguishable. Especially the definitions of interpretability and explainability are often used in similar contexts. The following chapter focuses on the terminologies and taxonomies found in the field of XAI. Additionally, it highlights common criteria and categories of explainability approaches such as post-hoc, ante-Hoc, self-explanatory, and global and local explanations.

6.3.1. Terminology

Chakraborty et al. [69] utilize the concept of explainability of evaluating the thoroughness of the model's output, which includes not only the prediction but also the reasoning behind it. In this context, thoroughness refers to whether all pertinent aspects of the input are included in the explanation. Additionally, they suggest using the term interpretability to evaluate the

quality of the explanation based on how easily it can be understood by humans. This definition coincides with that of Doshi-Velez & Kim, who define interpretability as “the ability to explain or to present in understandable terms to a human” [70]. Arrieta et al. [71] also gives definitions for explainability, interpretability, and also comprehensibility. These definitions are adapted within this thesis:

- **Comprehensibility:** refers to the ability of a learning algorithm to present its learned knowledge in a manner that is understood by humans. It is often directly related to the complexity of the model, therefore, the more complex the model becomes, the negative influence it has on the user’s understandability.
- **Interpretability:** is about the ability to explain or provide the underlying reasons or meanings behind the model's decisions in understandable terms to a human. It leans more towards articulating the reasons or logic behind the model's decisions, rather than just presenting the outcomes in an understandable fashion.
- **Explainability:** is linked to the concept of providing an explanation that serves as a bridge between humans and decision-making systems, allowing for a better understanding of the reasoning behind the model's decisions.

In essence, comprehensibility, interpretability, and explainabiity all aim to make complex models more understandable to humans. Comprehensibility focuses on presenting the model's learned knowledge in digestible chunks. Interpretability goes a step further by elucidating the underlying processes or reasons behind the model's decisions. Meanwhile, explainability serves as a bridge, presenting an interface that not only accurately represents the decision-making entity but also makes its decisions and processes comprehensible to humans, thereby integrating the aspects of both comprehensibility and interpretability for a more intuitive understanding and interaction [71]. These are just some of the various attempts that have been made to provide clear definitions for these terms. It should be noted that these definitions are often informal and lack a level of mathematical precision [72].

Explainable AI refers to the ability of an AI system to provide humans with clear and understandable explanations for its decisions and actions. This encompasses both "model explanation," which illuminates the overall workings of the AI system, and "decision explanation," which clarifies specific predictions made by the system. In this paper, "explainability" mainly refers to "decision explanation", focusing more on the rationale behind individual predictions instead of a comprehensive analysis of the whole model.

Several criteria are used in the topic of explainability in past literature [73]. These are often used for evaluating such systems. The definitions of some of the most relevant ones are as followed:

- **Robustness:** an explainability approach is considered robust if the explanation provides reliable and consistence explanations through variance is the data, like in the presence of noise, errors or adversarial attacks [74], [75].
- **Faithfulness:** is the ability to accurately represent the depiction of the underlying reasoning process behind the prediction of the model [20].

- **Plausibility:** refers to the ability to provide explanations that are persuasive to humans [20].
- **Understandability:** refers to the ability to illustrate the connection from the input to the output of the model in relation to the systems parameters. It is often defined as the user's cognitive conception of the model and the underlying functionality, the reasoning why a model has predicted a certain output or the ability to reason why a model failed in a particular task [13], [21], [22].
- **Satisfaction:** refers on the level of how well users perceive the understanding of the system that is being explained [24].
- **Sufficiency:** refers to the provision of adequate information to the end user to establish causation [23].
- **Trustfulness/Trustworthiness:** is a factor that is directly shaped by the interaction of the user with the system over time and through use. It affects the user's level of comfort when using the system. The perception of the person is directly linked to the beliefs of the user in the output of the system [21], [22].
- **Usefulness/Helpfulness:** in this thesis, usefulness is defined as the ability to provide the user explanation that helps to make the decision of the model more reasonable for the person [24], [21].

These terms can definitions are just a guideline to understand these terms. However, precise definitions and interpretations may vary based on the context of use and the specific research field [22].

Developing universally accepted taxonomies in the domain of XAI has been challenging due to inconsistent terminology and varied focus among different studies. While efforts to classify XAI concepts and methods exist, they often diverge notably in their terminology and classification categories, creating confusion both for experts and beginners in the XAI field [76]–[79]. Despite these issues, several criteria have been suggested to distinguish methods more effectively, as mentioned in [76] which are:

- **The functioning-based approach**: focuses on the underlying architecture and structure of the methods and how the XAI method can extract explanations/information from the model.
- **The results-based approach:** focuses on the output of the explainability method.
- **The conceptual approach:** distinguishes explainability methods due to conceptual dimensions and hierarchical levels.
- **The mixed approach:** which is a hybrid of the mentioned approaches above.

These approaches are not described in detail within this thesis. However, it is highly recommended to refer to [76] for more in-depth information. Based on taxonomies of previous literature, [76] combined several past approaches and proposed a new taxonomy which is shown in Fig. 13.

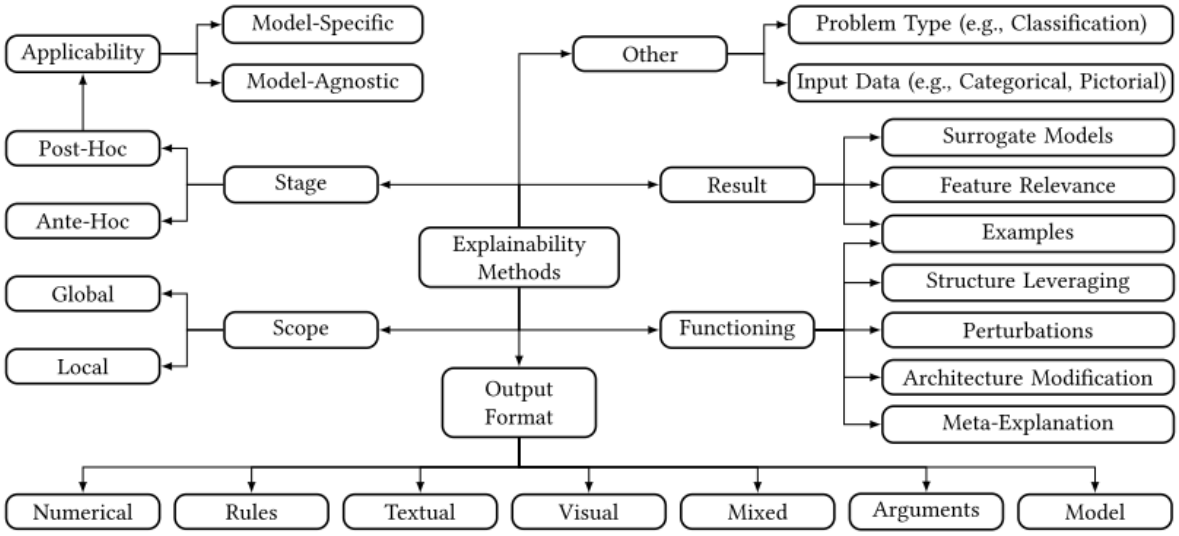


Fig. 13. Taxonomy of XAI methods. Adapted from [76].

The proposed taxonomy is built upon six main criteria:

- **Scope:** determines if the method in general and can be applied to any model or only to a specific type of model. The differences of global and local methods are described later in this section.
- **Stage:** whether the method is applied during the building stage (ante-hoc) or after the model is already build (post-hoc). The post-hoc category also includes model-specific and model-agnostic methods which are also described later in this section.
- **Output Format:** refers to in which format the output is presented.
- **Functioning:** focuses on how the model works. This could be by simplifying a complex model or by highlighting the importance of different features.
- **Result:** This criterion refers to the specific types of outputs generated by the explanation method. It could encompass prototypes that offer representative examples, heat maps that visualize attributions, or surrogate models that approximate the original model's behaviors, aiding in clearer interpretation and analysis.
- **Other:** the author also introduces a catch-all category for all the methods that do not clearly fit in the mentioned categories above [76].

Global & Local Explanations

Global explanations focus on the entire logic of a model, with the goal of understanding the inner workings for an input through the entire reasoning process of the model to the output. This category includes models that are designed to be explainable, such as Decision Trees and other rule-based systems that use algorithms that learn logical relationships between data [80]. For Decision Trees, the interpretable part could be the selected features and the cut-off points as well as the leaf node predictions. For linear models these parts could be the weights. In essence, while linear models offer some clarity in understanding features, each feature doesn't stand alone; it's often connected to others. These connections are clearer in linear models than in more complex ones like DNNss [81]. Local explanations

focus on the individual predictions made by the model, often used to explain how a model behaves in a certain scenario or for specific input instances. These local explanations are often used by users which are directly impacted by the decision of the model [82].

Post-hoc, Ante-Hoc and Self-Explanation

Post-hoc explainability methods are techniques used to analyse and interpret a ML model after it has been trained. They can be divided in model-agnostic and model-specific methods. Model-specific methods focus on the structures and parameters of the underlying models and are used in models which are often explainable by design. These methods are beneficial if the goal is to get a deeper insight and understanding of the decisions and the internal workings of the model. Model-agnostic explainability methods are independent of the underlying structure of the model and are usually applied on the output of the model. This is why they can be applied to any underlying model architecture. They can handle different feature representations like word embedding vectors for text models and the explanations can be adapted to specific use cases which makes them flexible. As mentioned, these methods are applied to the model's output, rather than the model itself, to gain a deeper understanding of the model's decision-making process. Some examples of post-hoc explainability methods include feature attribution, which examines the importance of different features in the model's predictions. Other techniques are example-based analysis, which looks at individual instances (examples) to understand how the model is making its predictions [9], [81], [82]. Another approach is the use of surrogate models, which are simpler, more reasonable models that are used to approximate the behavior of a more complex model. One common post-hoc explainability method based on the concept of surrogate models is LIME [9]. More information about that technique is provided in section 7.3.

Ante-hoc methods are often explainable by design aligning with “glass-box” approaches, where the internal workings of the model are transparent and easily interpretable. Common examples are Linear Regression or Decision Trees. Ante-hoc methods showed noticeable results because of their scalability and applicability in various domains including medicine, where unexpected data patterns are common. They still match the accuracies of more complex but less interpretable models [83].

Self-explaining methods, provide the explanation simultaneously with the prediction by utilizing the information generated by the model during the prediction process. Global self-explanatory models include Decision Trees and other rule-based models, while local self-explanatory models include for example feature saliency techniques using the models attention mechanism [84].

In this chapter, we delve into common terms and principles used in XAI. We found that this field suffers from inconsistent terminology and categorization of explainability approaches, which makes formulating effective taxonomies a challenging task. Moving forward, in the next section we will focus on recent developments in XAI in the field of NLP.

6.3.2. State-of-the-Art on Explainability for NLP Approaches

Neural networks for NLP are complex in their architecture which leads to better performance but poor transparency regarding the model decisions. Especially architectures like Transformers make it hard to resonate the model's decision by humans. When such networks are used for sensitive content where the decision of a model could have serious consequences, the understanding of why a model has made a certain decision is essential. Several recent approaches have attempted to implement methods to explain the decision of Transformers [13]-[16], [85]. In this section, we will examine the latest developments in XAI methodologies within the realm of NLP.

In the review of existing literature, special attention was directed towards exploring XAI approaches within the domain of NLP, with a particular focus on text classification. The subsequent table highlights a variety of advanced techniques widely recognized in the NLP field, each with an outline of its respective use case.

Table 1 State-of-the-art resources and papers on XAI for NLP approaches

Author	Year	Use Case	Model	XAI Method(s)	Ref.
Attanasio et al.	2022	Misogyny Detection	BERT	- Gradients [86] - Integrated Gradients (IG) [19] - SHapley Additive Explanations (SHAP) [87] - Sampling-And-Occlusion (SOC) [88]	[13]
Wu and Ong	2021	Sentiment Analysis	BERT	- Gradient Sensitivity (GS) [89] - Gradient*Input(GI) [90] - Layerwise Relevance Propagation (LRP) [91] - Layerwise Attention (LAT) [92], [93]	[14]
Bodria et al.	2020	Sentiment Analysis	BERT	- Integrated Gradients (IG) [19] - LIME [9] - Attention Weights [15]	[15]
Krishna et al.	2022	No specific Use Case	LSTM	- LIME [9] - KernelSHAP [87] - SmoothGrad [94] - Gradient*Input[95] - Integrated Gradients (IG) [19] - GradCAM [96]	[16]
Brasoveanu and Andonie al.	2020	Summary of XAI	BERT	- Subject focused visualizations:	[85]

		Approaches and Tools		○ relation between attention and model outputs [92], [97]–[99] ○ analysis of captured linguistic information via probing [99], [100] ○ interpretation of information interaction [101], [102] ○ multilingualism [100], [103] - Holistic visualisations: ○ BertViz [104]–[109] ○ Clark [105] ○ VisBERT [110] ○ ExBERT [107] ○ AttViz [108] ○ Kobayashi [109] - BERT Lang Street [111]	
Velampalli et al.	2023	Sentiment Analysis	SBERT and USE + FCNN and LSTM	- LIME [9]	[112]
Ansari et al.	2022	Hate Speech Detection	LSTM, CNN	- LIME [9] - Integrated Gradients (IG) [19]	[113]
Sebbaq and Faddouli	2023	Cognitive Text Classif.	MTBERT-Attention	- Own explainable classifer - LIME [9]	[114]
Mehta and Passi	2022	Hate Speech Detection	DT, LSTM, BERT + variants	- LIME [9]	[115]
Das et al.	2022	Propaganda Detection	BERT, BART, KNN-BART	- ProtoTEx [12]	[12]
Sourati et al.	2023	Local Fallacy Identification	Electra, BERT, DeBERTa, RoBERTa, DistilBERT	- ProtoTEx [12] - Case-based reasoning framework - Knowledge Injection Framework	[18]

* FCNN. = Fully Connected Neural Network

A hate speech related approach was done by Attanasio et al. [13]. They performed a post-hoc interpretability approach for Transformer-based detection of misogyny in tweets. In their paper, they focused on using a BERT Transformer for this task. They also compared four different feature attribution explainability-methods: Gradients [86] and Integrated Gradients [19], Shapley values-based methods (SHAP [87]) and Sampling-And-Occlusion (SOC [88]). They evaluated their experiments by plausibility and faithfulness as defined in Jacovi and Goldberg [20]. In their case, the definition of plausibility applies to if a highlighted words in a sentence can convince humans if the prediction of the classifier is misogyny or not. The faithfulness got evaluated by measuring the changes of the model prediction when erasing input tokens and observing if the model prediction changes. They claim that the explainability results of their gradient and attention-based methods are not faithful and that the gradient-approaches were inconsistent [13].
Similar results have been found in previous work [85], finding that LIME is more vulnerable to adversarial attacks than SHAP and concluding that the use of post hoc explainability methods like SHAP and LIME are not recommended for sensitive applications. However, their attention-based approaches did not provide any useful insights regarding their classification. Comparing to those, the explanations of the SHAP and the SOC approach were more faithful and plausible and therefore preferred for explaining their misogyny task than LIME or their attention-based method.

Wu and Ong [14] focused on comparing attribution methods like Gradient Sensitivity (GS) [89], Gradient*Input (GI) [116], [90], Layerwise Relevance Propagation (LRP) [91] and Layerwise Attention (LAT) [92], [93] for their sentiment classification task using BERT. They study the validity as well as the robustness of these four attribution methods. Since they do not explain their definition of validity, it is assumed that they mean the same as “faithfulness”. For evaluating the validity, they performed an Ablation study where words from text were removed successively and in a defined order. The relevance scores are compared after the changes. The experiments showed that performance and validity of the GS, GI and LRP changed similarly, whereas LAT was not suggested for retrieving relevance scores since it seemed to get distracted by irrelevant words.
The robustness of their model is evaluated by retraining their model under two different random initializations and by computing the relevance scores afterwards. Their results show that random initialisation does change the results of the model to a limited extend. They also mention that longer sentences suffer more from random initialisation than longer ones. They conducted experiments to see if the attribution methods work the same across different similar datasets. According to them, these experiments showed that the weights acquired in BERT could potentially be applicable to other tasks with comparable semantics.

Bodria et al. [15] trained a BERT Transformer for sentiment analysis and conducted explainability experiments using IG [19], LIME [9] and Attention Weights [15]. The explanation scores computed by the three approaches got compared with the ground truth labels and the BERT predictions. For their Attention Weight approach, they multiplied the attention scores (which are the output of the *softmax*) with the weights of the classifier to overcome that the resulting values are all positive. Fidelity [117] was applied to compare the explainability behavior of the black-box model to the XAI technique. Also, the similarity of their explanation scores to the ground truth label was measured by calculating ROC and

AUC scores. They claim das IG was the best method in their experiments but has high computational costs. However, comparing the computation time, LIME performed even worse. The attention-based method was the fastest even though it comes with a performance trade-off. Some of their experiments showed that the explanation scores contradict with the model predictions. It is essential to note that according to them, the agreement between the model predictions and the explanation scores are not always certain and the agreement in between the XAI techniques cannot always be guaranteed.

Brasoveanu and Andonie [85] provide a great overview of current state-of-the-art libraries/visualisation tools for explaining Transformers like BERT, mainly focusing of on attention-based approaches. They differentiate between subject focused visualization for visualizing just one specific aspect of the model and holistic visualizations which include tools used to explore the architecture of the models more in depth. Some, but not all mentioned libraries include: BertViz [104], Clark [105], VisBERT [106], ExBERT [107], ATTViz [108], and Kobayashi [109]. Due to them, future tools should focus on components like the embeddings, attention heads and the additional NNs of the Transformer as well as the used corpora. Also, summarizing the models' state through the architecture by using visualizations as averaged attention heatmaps as well as visualisations of the model states at various levels should be developed. The development of automatic and model specific visualizations for complex models is also mentioned as future work.

Krishna et al. [16] used tabular data, texts and images for comparing six post-hoc XAI methods using eight different models depending of the used dataset for prediction. For the textual dataset, an LSTM model was used. The focus of this study was to conduct a comparison to find out which method works the best, but also to tackle the problem of disagreement between explanations of the chosen techniques. The explanation methods used are perturbation-based (LIME [9], KernelSHAP [87]) and gradient-based (SmoothGrad [94], Gradient*Input[95], Integrated Gradients [19], GradCAM [96]). They organized user studies pursued by data scientists for the evaluation as well as additional statistical analyses. The results of the user studies as well as the heuristic evaluation showed conflicting explanations and disagreements comparing the explanation results of the different techniques. Additionally, they discovered that the participating data scientists frequently tended to choose their preferred explainability technique when asked which one worked better for them during the evaluation procedure. Some of the participants even reported experiencing this issue of disagreement in real-world applications, with a portion of them admitting to being unsure of how to address the problem.

Velampalli et al. [112] evaluated the performance of different AI models for sentiment analysis on datasets consisting text and emojis. They initialised Sentence-BERT (SBERT) and Universal Sentence Encoder (USE) for generating sentence embeddings and a standard fully connected NN as well as a LSTM model for the classification task. They afterwards used SHAP for explainability. They found that their LSTM model worked the best in both, the text, and the emoji dataset. The sentence embeddings generated from USE and SBERT improved the performance of the models. They used SHAP to validate if their models were discriminating or biases against users. They found that the SHAP explanation

were effective in identifying the strengths and weaknesses of the model. However, the authors did not mention any quantitative evaluation metric for the SHAP algorithm.

Ansari et al. [113] investigated in improving hate speech detection through data augmentation.
For their data augmentation task, they used easy data augmentation (EDA), Bidirectional Encoder Representations from Transformers (BERT) and back translation (BT). They used LIME and Integrated Gradients for explainability and measure their explainability techniques on random chosen test samples as well as on original and augmented data. The metrics applied were area over the perturbation curve (AOPC), log-odds, and coherence scores. They found that their augmented datasets improved the performances of their classification models which were a LSTM and a CNN. They also performed a post-hoc analysis of their models using the attribution scores of the explainability methods and found that they were useful in identifying strong features (features which resulted in a correct prediction of the model and neutral words that resulted in an incorrect predictions). A comparison of LIME and Integrated Gradients (IG) showed that Integrated Gradients tend to assign different attribution scores to a token if it occurs multiple times in a sentence depending on the surrounding context. LIME assigned the same attribution score to the same words regardless of the context. However, they did not make a recommendation which of the two methods is preferred. They claim that for future work, perturbation-based methods could be used for global model explanations.

Sebbaq and Faddouli [114] proposed a unique and explainable classification model called MTBERT-Attention which is based on BERT, multi-task learning (MTL) and the co-attention mechanism. The indent of the model is on cognitive text classification. They also investigated in an explainability framework based on the attention mechanism of the model which aims to provide explanations through the model's prediction. The found that adding a co-attention mechanism is beneficial not only for the classification task, but also for the explainability framework. The BERT tokenizer tends to split words into multiple sub-word-tokens which can be a problem for interpretability. The co-attention layer combines attention scores of split tokens, producing a consolidated score for the whole word.
To compare the explanation scores to the ground truth of their black-box model they trained a simple classifier using a $softmax$ activation function which aims to make predictions based on the explanation scores. They adapted LIME to fit their multi-task learning using it as a benchmark for evaluating their explainability framework. They used the fidelity metric (F) and a computational cost calculation to evaluate their results. They found that even though their explainability framework showed high fidelity and was computationally efficient, there were instances where tokens like adverbs or prepositions got high attention scores, making the attention mechanism challenging to interpret. These issues arise from the global nature of their explanations. In comparison, the LIME explanations focus on local explanations which might be more beneficial for a detailed insight for specific instances.

Mehta and Passi [115] discusses the use of XAI in detecting hate speech on social media. They conducted an in-depth literature research of several classification and explainability tasks.

The authors utilized two datasets and implemented different models, including Decision Trees (DT) and LSTM. The employed LIME and introduced variations of BERT (like BERT + MLP (Multi-Layer Perceptron) and BERT + ANN (Artificial Neural Network). They found that their BERT variants performed better than their linear explainable model. They used various measures from the ERASER benchmark to evaluate their results. BERT + MLP showed the best results in their experiments in plausibility according to the IOU (Intersection Over Union), F1-score, token F1-score, and AUPRC (Area Under the Precision-Recall Curve) metrics, indicating a convincing interpretation to humans.
Overall, the BERT variants demonstrated better performance and explainability compared to the linear models. They also evaluated the models on several bias metrics which are not further mentioned in this paper. Also, in this case the BERT variant models performed better in reducing unintended model bias for all these bias metrics compared to the other models. However, the researchers highlight the need for further investigation into reducing unintended model bias.
The authors highlighted the need of more diverse metrics to understand the model explanations better and that the impact of model performance on individual communities should be considered in further experiments. The study points out the difficulties when it comes to detecting hate speech, especially when sarcasm is present. The authors suggested to enhance in the identification of sarcastic elements in the texts which could lead to better performance of the models.

Das et al. [118] introduced a white-box classification architecture based on prototype networks as in Li et al. [119] that was first introduced for vision tasks. Their concept is based on generating prototype tensors from encoded latent clusters that are received from their used training samples. The underlying architecture is based on Transformer encoders. They experimented with the use case of propaganda detection in text and trained a ProtoTEx model with an underlying BART encoder (BART-large [120]) and compared it to a KNN-BART baseline and a BERT-large [7] black-box classification model. Their ProtoTEx model based on the BART model demonstrated equivalent performances to the BERT model and even slightly surpassed the performance of the KNN-BART baseline. Even though their ProtoTEx architecture is called a white-box model, according to their explanations received with the ProtoTEx model they claimed that their prototypes may not be beneficial for users if all prototypes are close to only a few training samples. An in-depth explanation of the functionality of ProtoTEx is provided in section 7.

Sourati et al. [18] adapted the ProtoTEx architecture of [118] for the use case of fallacy detection, coarse-grained and fine-grained classification. They also combined their techniques with approaches for data augmentation and curriculum learning. They did not only focus on prototypes, but also other concepts adapted from instance-based reasoning and knowledge-injection. BERT [85], They used several models for their classification experiments like DeBERTa [121], DistilBERT [122], Electra [123] and RoBERTa [124]. They adapted the Electra model [125] for their prototype -based experiments. In their experiments, they noted that their models had issues with understanding the broader concepts of the classes, which leaded to inconsistent prototypes, a varied relevance in instance-based examples and occasionally misleading results retrieved by their knowledge-

injection approach. They claimed that these results might be due to the challenging task of facility detection and recommend testing their methods in more realistic settings.

After this in-depth review of both the foundational concepts and recent developments in the field of XAI for text analysis and classification, we will delve deeper into a detailed discussion regarding the background and the most important aspects highlighted in the previous sections.

Discussion

The evaluation of various explainability approaches in recent years reveals a diverse range of strengths and shortcomings that are critical in choosing a XAI method depending on the underlying model and the specific use cases. A concise analysis of these methods, which serves as a basis for the selection of the method used within this thesis, is presented below:

- **Gradient-based methods:**
  Despite potential inconsistencies and vulnerability to threats, making them less ideal for critical tasks, they are emerging as notable alternatives to attention-based approaches. Integrated Gradients, offers a promising avenue for further exploration, despite noted issues regarding faithfulness and inconsistencies [13], [126], [85].
- **Attention-based methods:**
  These approaches, commonly found in models like BERT, use attention weights to focus more on the surrounding context than on the actual words themselves. This can cause a lack of detailed information showing how individual input elements (tokens) affect the predictions. They also encounter issues with being easily distracted by irrelevant words and inconsistent explanations [14], [127]. Enhancements to these methods are necessary to overcome issues of interpretability due to tokenization. The addition of a co-attention mechanism, has shown potential in addressing this issue by computing a score for the whole word instead of potentially generated subword tokens [114].
- **Perturbation-based methods:**
  While methods like LIME offer simplicity, they often lack context sensitivity [113]. This is a drawback that other techniques like IG tend to address more effectively. The selection of such methods should depend on the specific requirements of the use case, the problem and the model, balancing efficiency, performance, and interpretability [15], [16]. The SHAP and SOC methods have demonstrated a higher level of plausibility and faithfulness in their explanations compared to methods like LIME or attention-based techniques [13], [85]. Moreover, LIME and SHAP are both not recommended as a XAI methods for applications with sensitive content [85].
- **Prototype-based methods:**
  These methods, originated from vision tasks were utilized in studies by Das et al. [118] and Sourati et al. [18] employ prototype tensors generated from latent clusters in training samples for tasks such as propaganda and facility detection. Despite showcasing potential in terms of performance and explanation capabilities, they exhibit certain limitations. There's a notable tendency for prototypes to cluster close to a limited number of training samples, restricting their explanatory reach [118]. Since this

prototype architecture is functioning as a white-box model and has great potential for being used for making decisions of classification models more explainable, more investigation regarding this concept is needed.

Some further research directions and conclusions emerging from the literature review are:

- There is no one-size-fits-all method; the selection depends greatly on the specific requirements of the application and the necessary trade-offs [13], [14]–[16], [85], [113], [114], [126].
- Future developments should focus on creating sophisticated tools to better interpret and visualize complex models like Transformers, including a deeper exploration of attention and co-attention mechanisms [114].
- Developing robust post-hoc explainability methods and exploring new or alternative methods, potentially borrowed from other domains such as computer vision, should be a considered [118].

After taking a close look at the different types of XAI methods, finding the best method is not straightforward and depends a lot on the specific details of each case. The methods chosen for this study - LIME, Integrated Gradients, ProtoTEx, and GlobEnc - represent a varied mix of common and new XAI strategies. After the knowledge earned from the precious sections, in the next section 6.3.3 a more detailed insight in the concepts that are used in the field of XAI for Transformers is given, focusing on the four categories of the chosen XAI methods used for the comparative study of this thesis.

6.3.3. Concepts of Explainability for Transformers

A variety of different methods have been utilised in the past to provide explanations of text models.
Since Transformers have complex and non-transparent architectures, additional methods for providing explanations are needed. The focus in this section is on providing an overview on explainability techniques that can be applied on Transformer models. The research includes conducting several methods from many different sources like already ready-to go tools and other XAI methods which can be used but are not limited to Transformers [10] -[12], [75], [76], [80], [92], [96], [128], [129] -[12], [92], [96], [130].
A visual representation of the found concepts and methods that can be used for Transformer is shown in Fig. 14. five main concepts were found during the research for NLP tasks: gradient-based, perturbation-based, attention-based, prototypes and counterfactuals. Even though, counterfactuals can also be used for Transformer explainability [131], this concept is not considered within this thesis and is just mentioned in the illustration for completeness. The four chosen concepts are described in more detail within this chapter.

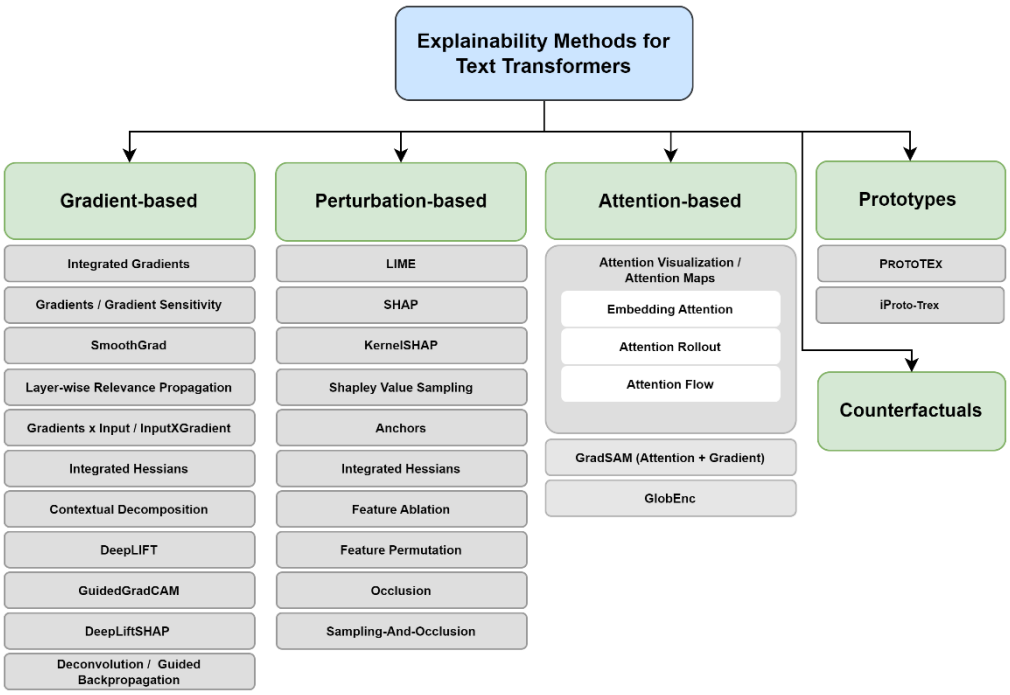


Fig. 14. Explainability methods for text Transformer based on [7], [10], [75], [76], [80], [92], [93], [94], [95], [96].

Perturbation-based

Perturbation-based methods, acknowledged as post-hoc and primarily local methods, involve modifying the model inputs and observing the changes in the predictions. Dropping words in a text as well as the use of adversarial examples for modifying the inputs so that the model is likely to misclassify the text are common approaches [87], [132], [133]. According to the survey of Ivanovs et al. [134], utilizing perturbation methods on text data is currently in the early stages of development. This might be due to the specific obstacles presented by text data, due to the individualized nature of words like meaning of the words in a sentence depending on its position and context. This XAI concept is more commonly used in other ML domains working with image data [134].

There are two ways to implement perturbation-based methods: feature omission and feature occlusion. In the context of explainable AI, omission means the concept of leaving out certain features of the input data (for example words in a sentence) when making predictions with a model. This strategy helps to understand the importance and significance of the selected features in the decision-making process of the model. It particularly allows to analyse how the absence of the missing feature influences the prediction. Occlusion involves hiding or masking parts of the input data to observe how the model responds to these changes. The contribution of a feature is positive when it pushed the prediction towards a certain class. The feature has a negative effect on the prediction when it diverts the prediction away from that class [135]. Removing different combinations of features could be beneficial to determine which part of the input affects the results the most. However, doing this for all possible feature combinations requires high computational power and is time consuming. A significant benefit of perturbation-based methods is their flexibility. Unlike many other methods that study the model as a

fixed entity, perturbation-based approaches can continuously interact with the model, forming and testing theories about it in real time. Additionally, they are versatile and can work with any model, no matter its structure. Being model-agnostic, they don not rely on the internal architecture of a model, allowing them to explain predictions from nearly all kinds of ML models including DNNs [134]. Utilizing perturbations on text data can be particularly challenging due to the distinct characteristics of words. Unlike image data where a minor change to a pixel generally doesn't affect a classifier's decision greatly, even slight adjustments to words can radically alter the overall meaning of a sentence, consequently having a substantial impact on the model's output [136]. One popular technique which is based on input perturbation is Local Interpretable Model-agnostic Explanations (LIME) [9]. This approach has been chosen for the experiments within this thesis. Therefore, more information on that LIME method can be found in section 7.3.

Gradient-based

Gradient-based approaches function as both local and global post-hoc methods which focus on the inner workings of complex NNs like Transformer models. They analyse the relationship between input features and predictions, either on individual instances (local) or across the dataset as a whole (global). These methods primary focus on examining the gradients of the model's output with respect to its input to evaluate the importance of each token in the input sequence [126]. Therefore, attribution scores are calculated via backpropagation from the gradients of the output (logits or *softmax* probabilities) to the extracted features of the input. More information on the backpropagation functionality is given in section 6.1.2. A variety of methods have been proposed to interpret models through the use of gradients [137]. However, since particularly complex deep learning models like Transformer models are non-linear, gradient-based approaches may not directly measure the effects of input perturbations. However, gradient-based approaches may not directly capture the multifaceted impacts of input perturbations. This limitation stems from the complex, non-linear relationships established between inputs and outputs in DNNs. In these networks, inputs undergo numerous transformations through layers of non-linear functions, resulting in a high-dimensional computational space where even minor changes can have a significant effect on the outputs. To better cope with the non-linearity, several variants of gradient-based explanation with modified backpropagation techniques have been proposed [138]. Examples of common gradient-based approaches are: Integrated Gradients (IG), Gradient*Input (GI), Layer-wise Relevance Propagation (LRP) and Gradient Sensitivity (GS) [93], [137]. One of the advantages of gradient-based methods is that they are fast since some of them (e.g., Gradient*Input) only require a single forward and backwards pass to calculate attribution maps. However, Integrated Gradients tends to be slower because more backward passes (50-200) are needed since computing the average gradient involves performing a numerical integration. But still, in comparison to other explanation methods it is still considered as efficient. Moreover, these methods scale relatively easily since the number of network evaluations, which are the processes of feeding an input through the network to obtain an output, are independent of the number of input features. Gradients tend to be noisy which can lead to attribution maps that appear to focus on irrelevant features, this is also likely due to the non-linearity and the high-dimensional representations of the features (e.g., in high-frequency variational pixels in image classification or word embeddings in large language models)

[138]. Integrated Gradients has been chosen as the gradient-based method within this thesis because it is robust as it provides a more stable and accurate attribution of feature importance by averaging gradients over a series of inputs, reducing the influence of complex, nonlinear relationships in the model. The method is explained in more detail in section 7.4.

Attention-based

Since Transformers are based on an attention mechanism, some studies using attention-based models, typically operating as local and global post-hoc methods, visualize the attention weights for interpretability directly [104]. Considering these attention weights as relevancy scores is often done in single attention layers where the attention heads of the Transformers are analysed. With these scores, the most relevant part of an input sentence that led to a certain prediction of the model can be highlighted. Simply visualizing the attention weights is one approach for explainability. Additionally, it is possible to highlight the attention scores not just only from the top-most attention heads, but also by using the attention scores from the underlying layers and heads. Clark et al., 2019 [105] found that visualizing the attention scores at different layers reveals linguistic correlations within the layers. Another technique to obtain such scores is to combine multiple layers. However, when the attention scores for each token get averaged, the obtained output will be blurry, and the role of the different layers are lost. An alternative approach is the rollout method, where attention points get reallocated by examining pairwise attentions. It is assumed that the attentions in the subsequent contexts can be combined linearly. This approach often gives better results than just using one single attention layer for generating explanation scores. However, this techniques also has its issues since it tends to highlight irrelevant tokens [68]. Also, research has shown that gradient-based approaches can often provide more accurate explanations, as opposed to methods based on attention mechanisms [126]. A more detailed information on the attention mechanism is provided in section 6.1.3 and 6.1.4. The attention-based method used within this thesis is GlobEnc [11] and is described in section 7.6.

Prototype-based

The literature review showed that the research regarding the use of prototype tensors, primarily functioning as global ante-hoc methods, for text classification explainability is rare [12], [130]. The concept of prototype classification is a traditional approach that is based on case-based reasoning, which is based on solving problems based on similar past problems [139]. The idea of prototypes, when thinking on a classification problem, is on creating representative examples for each provided class in the dataset. Therefore, data points are categorized according to how closely they resemble a representative or model example within the dataset. In this context, a prototype is similar to a data point (sentence) in the training set. Collectively, these prototypes aim to offer a good representation of the entire dataset. However, in some scenarios the authors of [119] claim that prototypes do not necessarily have to match a specific example from the training data but could be a combined representation of multiple data points. The chosen methods in this paper is called ProtoTEx [12], which builds upon the white-box classification method of Li et al. [119], enhancing it with the use of large-scale pre-trained language Transformers. More detailed information on this approach can be found in

section 7.5. Since the predictions of the model are based on the prototypes which are derived directly from similar examples in the training example, this model enables predictions to be faithful during the inference process.

In the sections we've covered so far, we took a deep dive into the approaches of text classification and ML. We started in section 6 with the basics of text classification, going through both traditional and more recent deep learning approaches. We also delved into the mathematical foundations of attention mechanisms. We then focused on the initial Transformer architecture and the structure of the BERT model in section 6.2, which a special focus on the pre-training and fine-tuning techniques. This led us to section 6.3 were we opened the conversation about XAI techniques in NLP, focusing on the terms and latest trends particularly in the field of text classification. A closer look on how Transformers can be explained, discussing main concepts like perturbation-based, gradient-based, attention-based and methods based on prototypes in detail is given in section 6.3.3. In the next section, we will narrow down our focus to explore the selected XAI methods chosen for the comparative study of this thesis.

7. Methodology

Building on the theoretical background, discussions and analyses in the preceding sections, the study centres around leveraging the BERT Transformer for text classification tasks. This choice is grounded in BERT's advanced capabilities in understanding language semantics. The goal is to choose one method of each of the mentioned concepts mentioned in section 6.3.3 to ensure an objective analysis and comparison of the chosen methods. The leveraged pre-trained BERT model and the selection of the chosen XAI methods is stated in this section.

7.1. BERT

For the experiments, a BERT Transformer, namely the bert-base-uncased model from Huggingface [140] has been fine-tuned on the three datasets described in section 8.2. This BERT Transformer is an English model, which got initially pre-trained using a Masked Language Modelling (MLM) and a Next Sentence Prediction (NSP) objective as described in section 6.2. This model is not case sensitive, which means that it makes no difference if a word is written in upper or lower case. This model also removes accent markers from the input texts. The initially pretrained bert-base-uncased model consist for 12 layer, 768-hidden layers, 12 heads and a total of about 110M parameters.

7.2. Selected XAI Methods

Depending on the previous state-of-the-art analysis and the introduction of different XAI concept, four XAI methods have been chosen. The concise reasons behind selecting them for the comparative study of the thesis are:

- **LIME [9]:** is the perturbation-based method. Despite its known shortcomings, it remains a simple and interpretable, and well-established choice suitable as a baseline [13], [85]. Its perturbation-based approach can offer valuable insights in cases where simplicity and interpretability are more critical, possibly acting as a benchmark for the newer, more sophisticated methods.
- **Integrated Gradients (IG) [10]:** is the gradient-based method. This method is chosen for its potential to overcome some drawbacks found in other gradient-based methods, offering more comprehensive analysis. While its faithfulness and consistency have been critiqued, the approach holds promise for further exploration since unlike some other gradient-based methods, it considers the entire path of changes in inputs, allowing for a more nuanced understanding of the model's behavior [137], [138].
- **ProtoTEx [12]:** is the prototype-based method. A response to observed need for new or alternative methods for post-hoc explainability, ProtoTEx's prototype-based approach offers a new perspective. Prototypes in ProtoTEx are representative

examples extracted from the training data. These prototypes serve as key reference points that the model uses to understand and explain its predictions.
This aligns with the aim of seeking fresh concepts, which might be adapted from different domains, to improve the model's interpretability.

- **GlobEnc [11]:** is the attention-based method. As an enhancement to attention-based methods, it aims to rectify the shortcomings found in traditional attention-based methods by integrating various elements of the encoder block in its explanation strategy. This choice was made because of the need of sophisticated tools that can provide a more detailed attribution analysis, especially improving interpretability in Transformer models, aligning with future directions highlighted in the literature review.

After explaining the selection of the four XAI methods building on the theoretical background, discussions and analyses in the preceding sections, the study centres around leveraging the BERT Transformer for text classification tasks. This choice is grounded in BERT's advanced capabilities in understanding language semantics. After examining various explainability concepts which were discussed in previous sections, I have selected four XAI methods for our study: LIME [9], which is based on input perturbations, Integrated Gradients [10], which is a representative of gradient-based approaches, a prototype-based approach called ProtoTEx [12], and GlobEnc [11], which represents an attention-based approach. The technical background of the selected method to get an understanding of their functionality and how they assist in making the BERT Transformer more transparent in text classification tasks is discussed in the upcoming sections.

7.3. LIME

One of the most popular techniques using input perturbation is Local Interpretable Model-agnostic Explanations (LIME) [9]. LIME creates a local surrogate model to explain the predictions of the original model. It does this by perturbing the input data and observing the changes in predictions. The local model tries to approximate the predictions of the original model as closely as possible within a local neighborhood around the considered data point. In Fig. 15., the global, complex model and its non-linear decision boundaries are illustrated.

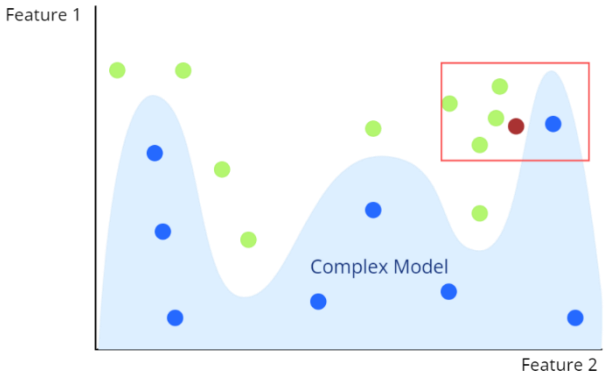


Fig. 15. LIME – Global model. Adapted from [90].

LIME utilizes locally linear models that are constructed around the predictions of the complex model. These linear models, as illustrated in Fig. 16. LIME – Local model. Adapted from [90]., are the so called “surrogate models”. Due to their simple and linear architecture, they are considered as easily explainable models since in such models, the effect of a variable is directly dependent of another variable. These surrogate models aim to mimic the classifiers predictions to explain the decision-making process of the complex model by identifying the specific input features that are driving the predictions depending on where the datapoint is in the decision boundary.

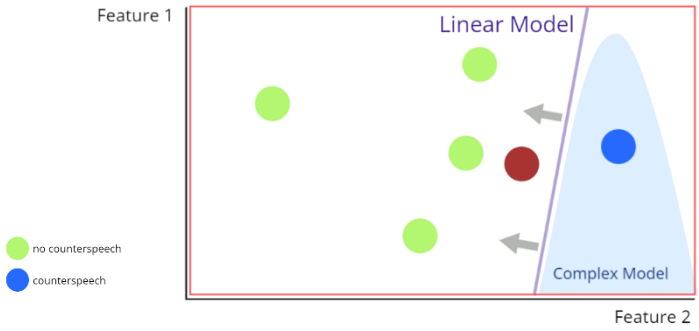


Fig. 16. LIME – Local model. Adapted from [90].

LIME produces new datapoints around the input datapoints, also called “perturbations” which are weighted to the distance of the input data point as shown in Fig. 17. LIME - Data perturbations. Adapted from [90]. These weights are needed since just the local points around the input data are important for the explanation. The perturbations can be achieved by a normal distribution with the mean and standard variation for each of the features. Afterwards, the predictions for the perturbations are made by the complex model, in our case the Transformer.

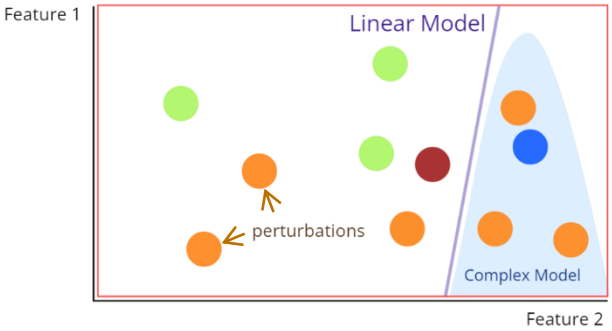


Fig. 17. LIME - Data perturbations. Adapted from [90].

By that we get the labels for each of the new datapoints. Using that data, a linear interpretable model can be trained. The calculated loss during training is also dependent on the distance of the new data points to the input point by a given weight. The more important the data point is, the higher is its weight. This dependency is visualised as a simple heat map in Fig. 18.

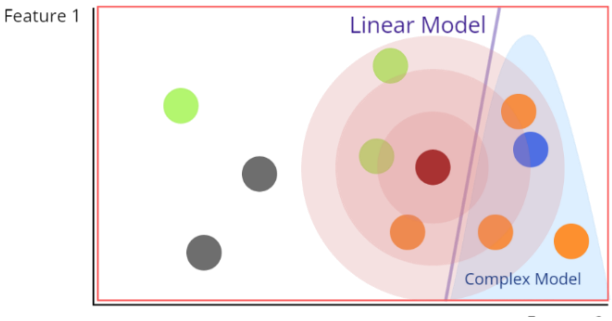


Fig. 18. LIME - Distance of the data points.
Adapted from [90].

For keeping the model as simple as possible, for the loss, an additional sparse linear model is used to produce as many serial weights as possible to ensure its simplicity. The importance of each word in a sentence can be observed by attribution scores or in this case, the prediction probabilities of the features or tokens (as in textual data). These can be visualised using saliency-/heat-maps. Since the explanations provided by LIME are local, they are considered as locally faithful but not necessarily globally faithful, since the focus is only on the local surrounding of the prediction. Also, prior knowledge of the topic can create more trust since humans can validate the explanation [9] .

LIME is computationally expensive, and the surrogate model is depended on the set parameters which can highly influence the results of the model and the needed computational capacities. However, LIME is model-agnostic and can be therefore used to explain the decision of any underlying ML model. It can be easily implemented, is flexible and easy to use, which makes it a versatile tool. Also, since LIME can generate local explanations by creating a surrogate model, it can be used for generating explanations for specific features/data points. LIME is sensitive to input perturbations since it needs to perturb instances by definition to generate a dataset of similar instances, which makes the method less robust to input changes [141].

7.4. Integrated Gradients

As mentioned in section 6.3.2, gradients-based methods suffer from a break in their sensitivity (how much of the output changes with respect the small input changes), since this metric relies on the assumption that the output of the model is a linear function. Depending on the model, these linear relationships are not always given [57]. However, Integrated Gradients relies on a modified version of calculating the backpropagation which makes the method more capable of dealing with this issue.
In Integrated Gradients, the input features are calculated by accumulating the gradients through the models path integral from the baseline to the output. The baseline is defined as the input without the presence of a particular feature which in theory should lead to a neutral prediction [19], [142].

To address the non-linearity inherent in DNNs, a distinctive strategy is utilized during backpropagation. Specifically, to calculate the attribution values via Integrated Gradients,

the baseline undergoes gradual transformation towards the input of interest, with one token being added sequentially at each stage. During each of these steps, the gradients (which are the partial derivatives of the output with respect to each input feature) are computed. These gradients represent how much each feature is contributing to the change in the model's prediction at each step. Instead of simply using the gradients at the final input, these gradients are averaged (integrated) across all the steps. This integration is done to capture the overall contribution of each feature across the entire path from the baseline to the input of interest which results in the desired attribution values. The attribution values represent the contribution of each input feature to the difference in the output between the baseline and the input of interest now can be used (e.g., visualized) to understand the model's prediction for the input of interest in terms of its input features [19].

Integrated Gradients tends to be slower than other gradient based methods (e.g., Gradients x Input, LRP) since it requires more backward passes (30-200). As mentioned above, computing the average gradient involves performing a numerical integration. But still, in comparison to other explanation methods (e.g., LIME), it is still considered as efficient [138]. Moreover, gradient-based methods in general scale relatively easily since the number of network evaluations, which are the processes of feeding an input through the network to obtain an output, are independent of the number of input features. Also, Integrated Gradients is model-agnostic, and can therefore be applied to any model and is easy to use. Even though IG tackles the problem of nonlinearity issue, the strategy does not completely resolve it. Therefore, due to the non-linearity as well as the high-dimensional representations of features (e.g., in high-frequency variational pixels in image classification or word embeddings in large language models) IG as well as other gradient-based methods tend to be noisy which can lead to attribution maps that may focus on irrelevant features [138].

7.5. ProtoTEx

Das et al. [12] introduce ProtoTEx, a classification architecture which uses prototype tensors for explaining the decisions of an NLP model. In their approach they integrate encoders from pretrained language models on top of the prototype classification network based on the implementation of Li et al. [119]. Their architecture of the ProtoTEx model consist of an encoder and an additionally added linear prototype layer. Within this layer, individual units hold weight vectors that bear similarities to prototypical examples [18]. The prototype layer consists of positive and negative prototypes, which are designed to aid models in differentiating between the presence and absence of features that contribute to a particular class. The network learns to create prototype tensors which represent latent clusters of training examples chosen by the model which are similar to the given input sentence. Since the classification itself is done by the linear layer which takes the distances to the prototype tensors as input, the network can be considered as a white-box model since the global explanations are directly linked to clusters that were learned from the training data [12].
The training process begins with the initialization of a set of prototypes, which are representative examples that the model will learn from. These instances are first put into an encoder which can be of any kind of architecture, to transform the input into a latent

representation. This representation shares the same space as the input data and the prototype layer. For each prototype j, the prototype layer computes the $L2$ distance (also known as Euclidean distance) between its representation p_j and the input x_i, i.e., $|| x_i - p_j ||_2^2$. This results in a distance vector (matrix of these $L2$ distances).
Additionally, a distance mask layer is used to mask the distance vector for the purpose of guiding the model to optimize the closeness of input examples to a specific set of prototypes that belong to the particular class. The masked distance vectors are then passed through a fully connected layer and a $softmax$ layer to classify a data point. To ensure that the prototype vectors are interpretable, the model is optimized by using additional losses explained later in this section. The linear layer learns a weight matrix of dimension $K \times m$ for K classes and m prototypes. The weights learned for each prototype indicate that prototype's relative affinity to each of the K classes. Classification is then performed via a $softmax$ function, which is a type of function that can convert a vector of arbitrary real-valued scores into a vector of probabilities.

The total loss is a weighted sum of three terms:

$$L = L_{ce} + \lambda_1 L_{p1} + \lambda_1 L_{p2} \quad (17)$$

In this formular, λ_s are hyperparameters, L_{ce} is the standard classification cross-entropy loss, and L_{p1} and L_{p2} are the two additional auxiliary prototype loss terms. The auxiliary loss terms are introduced to maintain the interpretability of the prototype vectors. L_{p1} minimizes the average squared distance ($L1$ norm, also known as Manhattan distance) between each of the m prototypes and at least one encoded input. L_{p2} ($L2$ norm, also known as Euclidean distance) encourages training examples to cluster around prototypes in the latent space by minimizing the average squared distance ($L2$ norm) between every encoded input and at least one prototype. These additional loss terms, aid a minor role in the model's optimization. A unique aspect of the ProtoTEx model is the use of an interleaved training procedure. This procedure alternates between optimizing the model's parameters and updating the prototype representations. This iterative process helps balance competing loss terms, encouraging each learned prototype to be similar to at least one training example (L_{p1}) and encouraging training examples to cluster around prototypes (L_{p2}). To encourage segregation among the prototypes, instance normalization is performed for all distances. This ensures that the prototypes represent more subtle patterns within the training examples belonging to the same class.

As already mentioned, a benefit of the model is that it provides global explanations directly linked to the learned clusters of the training data during inference, thereby functioning as a white-box model. Moreover, its case-based reasoning strategy lends itself to being considered as faithful [12], [18].

7.6. GlobEnc

The GlobEnc [11] method, is a novel approach to global token attribution analysis in Transformer based models. This method is designed to incorporate all components of the encoder block and aggregates this information throughout the layers of the model. GlobEnc

is based on the output of the encoding layer of the underlying model and includes the second layer normalization in the norm-based analysis of each encoding layer. To aggregate attributions over all layers, the method applies a modified attention rollout technique, returning global scores. This approach significantly improves this method over existing techniques for quantifying global token attributions.

The proposed method enhances the norm-based analysis technique introduced by Kobayashi et al. [109]. In their approach, RNES (Fig. 20) is the method for analysing the residual connection of the attention block which includes the attention block's Layer Normalization (LN#1) and the Residual Connection (RES#1). However, the encoders feedforward layers, RES#2 and the output LN#2 were not considered in their approach. R_{NESLN} refers to the degree of impact that the input token j has on its output token i in the encoder layer.
Since the encoder layer of a BERT Transformer consists of multiple components, the attribution analysis method GlobEnc (N_{ENC}) considers almost every of those for computing the aggregated attribution scores. So, in GlobEnc, the encoder layer components got included additionally from the attention block outputs $(\widetilde{z})_i$ up to the output representation $\widetilde{(x_i)}$.

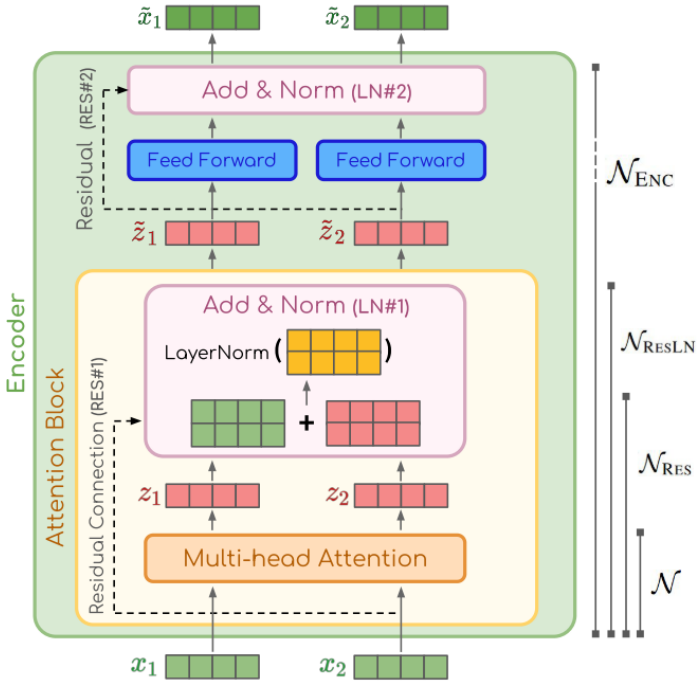


Fig. 20. Components that are included in each of the proposed token attribution analysis method within a Transformer encoder layer. The proposed GlobEnc method integrates the entire encoder layer (N_{ENC}). Adapted from [11].

For computing the multilayer attention, the layer wise analysis methods where aggregated by slightly modifying the rollout technique [92]. Where A is the attention map and l are the layers. In this method, all residual connections and multi-head attentions are assumed as

contributing equally. The attention rollout of layer l with respect to the inputs is computed using Eq. (18):

$$\tilde{A}_l = \begin{cases} \hat{A}_l \tilde{A}_{l-1} & l > 1 \\ \hat{A}_l & l = 1 \end{cases} \quad (18)$$

$$\hat{A}_l = 0.5\bar{A}_l + 0.5I \quad (19)$$

In GlobEnc, raw attention map over all heads in layer $\bar{A}_l$ got replaced with the desired attribution matrix in layer l. For methods without residual connection a fixed residual effect is used (19)

(5)(6) ($r_i \approx 0.5$) for calculating the aggregated attribution scores (19)

(5)(6).
The GlobEnc method has been shown to produce faithful and meaningful results, with high correlations with gradient-based methods in global settings. Also, the authors claim that norm-based methods achieve higher correlations than weight-based methods, incorporating residual connections plays an essential role in token attribution, considering the two-layer normalizations improve the analysis only if coupled together, and aggregation across layers is crucial for an accurate whole-model attribution analysis. The authors claim that their GlobEnc model produces meaningful results with high correlations to gradient-based methods. They mention their method to be faithful and robust. Even though the method needs to aggregate the attributions over all layers of the model, the needed computational time is rather fast.

In this section we have explored the four chosen explainability methods for the comparative study to enhance our understanding of machine learning model decisions. Moving to the next section 8, I will describe how the chosen XAI methods have been implemented in the real-world scenario of counter and hate speech detection. In this following chapter, an in-depth description of the utilized datasets and the training strategies of the models is given. This chapter also includes the evaluation strategies of the classification models as well as the evaluation procedure of the XAI models.

8. Experimental Setup

In this section, the use case of the comparative study is stated at the beginning, followed by a description of the identified datasets. Subsequently, we delve into the training procedure of the classification models, highlighting the application of BERT Transformer models for text classification. Afterwards, the evaluation approach of to assess the performances of the classification models as well as the evaluation procedure of the XAI methods are described.

8.1. Use Case

In the ever-evolving landscape of online communication, hate speech (HS) and its counterpart, counter (CS) speech, have emerged as significant areas of research and intervention. This thesis resonates with the “Counter Speech: Young People Against Online Hate” project[1], emphasizing the critical role of counter speech in combating the pervasive issue of online hate speech among youth. By fostering active participation in counter speech, the project aims to transform the online space into a more respectful and safer environment. Since many countries are implementing laws to reduce the amount of hatred found on the internet, social media platforms like Facebook and Twitter introduce monitoring systems to combat such content [3]. If hateful content is found, simply banning or deleting such comments is controversial since each deletion also restricts a person's freedom of expression to a certain extent [1]. Benesch [143] also states that censorship and punishment can cause inflammation regarding the conversation making the situation even worse.

Alternatively, to deleting comments, the use of counter speech also referred as “counter narrative” has shown great affect when it comes to combat online hate [25], [104]–[108]. By counter speech, the comment can be directly managed by the users independently from the internal monitoring system of a social media site. Thus, the users themselves act as moderators. This approach can be beneficial since automatic methods often need some time to find and categorise controversial input. Also, in some platforms, content needs to be reported first to get checked by monitoring staff. In conclusion, counter speakers can act more quickly to response to such content than the proposed moderation techniques. However, according to [144], several parameters have to be considered when measuring the effects of counter speech. One of those parameters is the proportional size of the group of counter speakers to the group of hate speakers [145]. Also, just like the size of the group of hateful speakers, the tone of the counter speech, the number of people which are in the conversation and who the audience is also influence the success of counter speech [146], [147].
Even though [148] stated that in their experiments on responding with counter speech to hateful postings did not stop haters from posting hateful content, it still had a positive effect in the means of reaching a larger audience to encourage more counter speech in general. However, Schieb and Preuss [145] found in their research that counter speech had indeed an effect on the original, hateful speaker. Moreover, there is not only counter speech with positive sentiment found on the internet, but also counter speech with negative sentiment as introduced in some papers. The authors of [4] is referring this type of speech "counter hate". Even though counter hate might not ease the inflamed situation, detecting such counter hate can still bring some benefits. For example, users who engage in counter hate still have tendencies to combat the hate in the first place. Such persons could be identified in social media platforms and subsequently encouraged to engage in more and/or appropriate counter speech. Building upon the foundational understanding of HS and CS, this thesis aims to bridge the gap between the theory and real-life interventions by focusing on the automatic detection of such content on social media platforms. The main use case

[1] https://research.fhstp.ac.at/en/projects/counterspeech-young-people-against-online-hate

of this work, therefore, pivots to the development of tools capable of identifying and categorizing HS/CS content. This initiative not only seeks to enhance the visibility and effectiveness of counter speech but also serves as a vital step towards fostering a safer and more inclusive online environment by encouraging more users to actively oppose hateful content. This highlights the urgent need for automated tools to assist in the quick recognition and promotion of counter speech, laying the groundwork for a more proactive and educated strategy to combat online hate speech. After explaining the use case of this thesis, we will move forward to the next chapter exploring the considered datasets for training the classification models.

8.2. Datasets

To carry out a comprehensive study, we require annotated data. Crafting a new dataset from scratch is not only complex but also falls beyond the scope of this thesis. Therefore, a thorough literature search was undertaken to identify existing datasets suitable for the analysis of HS/CS. Specifically, three particularly promising datasets were uncovered, which are outlined in Table 2. For this research, datasets were selected that help to identify CS in two specific situations: comparing CS against non-CS and comparing CS (which can also include hate) against HS (which supports a hateful comment in a hateful way).

For the study, three datasets were selected: "Thou Shalt Not Hate" [25] and "HateCounter" [26], which are related to counter speech, and the language classification dataset "Europarl" [149]. The Europarl dataset was chosen specifically for evaluating the ProtoTEx approach because of its large size and the relative simplicity of the tasks it involves. This makes it an excellent baseline for our study, as it can potentially yield high-quality solutions. In contrast, the tasks in the other datasets are more complex, which might restrict the effectiveness of the model's explainability. Utilizing Europarl as a baseline helps to mitigate any negative impacts on explainability that might arise with the use of the more complex datasets, ensuring that our model maintains a high degree of accuracy and effectiveness throughout the study.

Table 2 Overview of the used datasets

Dataset Name	Labels	Platform	Number Samples	Ref.
Thou Shalt Not Hate	CS, non-CS	Youtube	13,924 comments	[25]
HateCounter	HS, CS	Twitter	1,290 HS-CS pairs 223 HS-HS pairs	[26]
Europarl	21 European languages	Proceedings of european parliament	~ 2M sentences per language (English & German)	[149]

8.2.1. Thou Shalt Not Hate Dataset

The Thou Shalt Not Hate dataset [25] is an English dataset which includes comments from Youtube. The dataset has one counter speech (CS) and one non-counter speech (non-CS) class. The class distribution is relatively balanced with 7,024 samples for counter speech and 6,898 for non-counter speech. This makes a total of 13,922 samples. The distribution of the dataset splits can be found in Table 3.

Table 3 Dataset splits of the Thou Shalt Not Hate Dataset

	Training Split	**Validation Split**	**Test Split**
Full Split Size	8,909 samples	2,228 samples	2,785 samples
Split Size Class 0 (CS)	4,426 samples	1,109 samples	1,363 samples
Split Size Class 1 (non-CS)	4,483 samples	1,119 samples	1,422 samples

The resulting training dataset had a size of 8,909 samples, the validation dataset a size of 2,228 samples and the test dataset a size of 2,785 samples. The test set underwent additional pre-processing since it was also used for the user study. For more details on the pre-processing are provided in section 8.4.3. Therefore, links and user references were replaced with “<LINK>” and “<USER>” for the human evaluation and the ablation study. An insight in some examples of the dataset is given in the following table:

Table 4 Example texts of TSNH dataset

Nr.	Class	Text
1	0 (CS)	it's illegal for African Americans to sit down now?
2	1 (non-CS)	The girl hardly said anything or flinched rather than back him up!
3	0 (CS)	My boyfriend is Jewish and I'm an atheist and German. We get a shit load of remarks. But if anyone said something like that to him, I would slap them so hard, I swear...
4	0 (CS)	Unnatural? Lol, he obviously hasn't studied any biology... Homosexuality is ubiquitous throughout nature
5	1 (non-CS)	i don't know why Christians are shocked. i am guessing christians don't read their bible.

8.2.2. HateCounter Dataset

The HateCounter dataset [26] consists of 1,290 hate speech - counter speech pairs (the starting point is hate speech and the response is counter speech which can also be hateful) and 223 hate speech - hate speech pairs (were the starting point is hate speech and the answer is supporting hate speech). In this work, only the responses were used for the experiments. The splits of the datasets can be found in Table 5**Table 5** Dataset splits of the HateCounter Dataset.

Table 5 Dataset splits of the HateCounter Dataset

	Training Split	Validation Split	Test Split
Full Split Size	1,059 samples	302 samples	152 samples
Split Size Class 0 (CS)	903 samples	257 samples	130 samples
Split Size Class 1 (HS)	156 samples	45 samples	22 sample

The splits result in a training dataset size of 1,059 samples, a validation dataset size of 302 samples and a test dataset size of 152 samples. Links and user references were replaced with “<LINK>” and “<USER>” for the human evaluation and the ablation study.

Some examples of the dataset are displayed here:

Table 6 Example texts of HateCounter dataset

Nr.	Class	Text
1	1 (CS)	Read the first sentence to yourself slowly
2	1 (CS)	@user You are a bitch you know that damn one day karmas gonna come back 3x harder and I hope it hurts like a bitch you fucker
3	1 (CS)	@user Twitter please don't let satan have a Twitter account
4	0 (HS)	bitch wat am i
5	0 (HS)	I'm a left sjw and i can't agree with you more!

8.2.3. Europarl Dataset

The Europarl dataset [149] is a summary of proceedings of the European parliament including 21 European languages. In this thesis, the dataset was simplified to a binary and balanced dataset with only German (DE) and English (EN) texts to obtain a maximally easy classification dataset to assess the baseline performance of the selected explainability methods. On the dataset homepage [150], several versions of the parallel corpus are available. In this thesis, version 7 of the parallel corpus including the languages German-English was considered. To make sure to get a dataset on sentence level, the texts were then transformed into one sentence per line which resulted in a German dataset of 2.111.448 sentences and an English one of 1.948.874 sentences. For the German dataset, the first 15.000 were considered and the sentences 15.000 - 30.000 were used for the English dataset. This step is taken to ensure that the sentences in both splits contain distinct information, as they share the same content but are written in two different languages. After shuffling both, each of the German sentences got labelled as “0” and the English ones as “1”. The splits sizes are shown in Table 7.

Table 7 Dataset Splits of the Europarl Dataset

	Training Split	Validation Split	Test Split
Full Split Size	21,600 samples	5,400 samples	3,000 samples
Split Size Class 0 (DE)	10,858 samples	2,734 samples	1,524 samples
Split Size Class 1 (EN)	10,742 samples	2,666 samples	1,476 samples

A selection of some samples of the dataset are displayed in the following table:

Table 8 Example texts of Europarl dataset

Nr.	Class	Text
1	1 (EN)	In contrast to how things are done here in the European Parliament and in Parliament' s committees, where rapporteurs are nominated and appointed to produce reports, there are no regulations whatsoever applied within this delegation to the Joint Assembly
2	0 (DE)	Unsere Vorschläge betreffen vor allem das Alter der Öltanker
3	0 (DE)	Ich bin dankbar dafür, daß die Kommission ein Aktionsprogramm angenommen hat.
4	1 (EN)	This is already happening.
5	1 (EN)	Moving onto another point: the Commission has passed a long list of its intentions to Parliament.

On each of the three selected datasets, BERT Transformers got fine-tuned to the specific classification task. In the following chapter, the training strategies and the resulting Transformer models are described.

8.3. Training Classification Models

BERT Transformer models can be trained in two fashions, including unsupervised pre-training and supervised task-specific fine-tuning. A detailed description about how these training strategies work is given in section 6.2. For the experiments in this thesis, an already pre-trained BERT model (see section 7.1) is employed and adapted to the dataset-specific tasks by fine-tuning them.

Each of the models followed the same fine-tuning objective using the respective dataset of their task, namely the Thou Shalt Not Hate dataset, which fine-tuned model is referred as TSNH-BERT, the HateCounter Dataset which model is named HC-BERT and the Europarl dataset, which fine-tuned model is called EP-BERT. In **Fig. 21**the process of adapting the pre-trained BERT model for creating the three task-specific classification BERT models using the mentioned datasets is displayed.

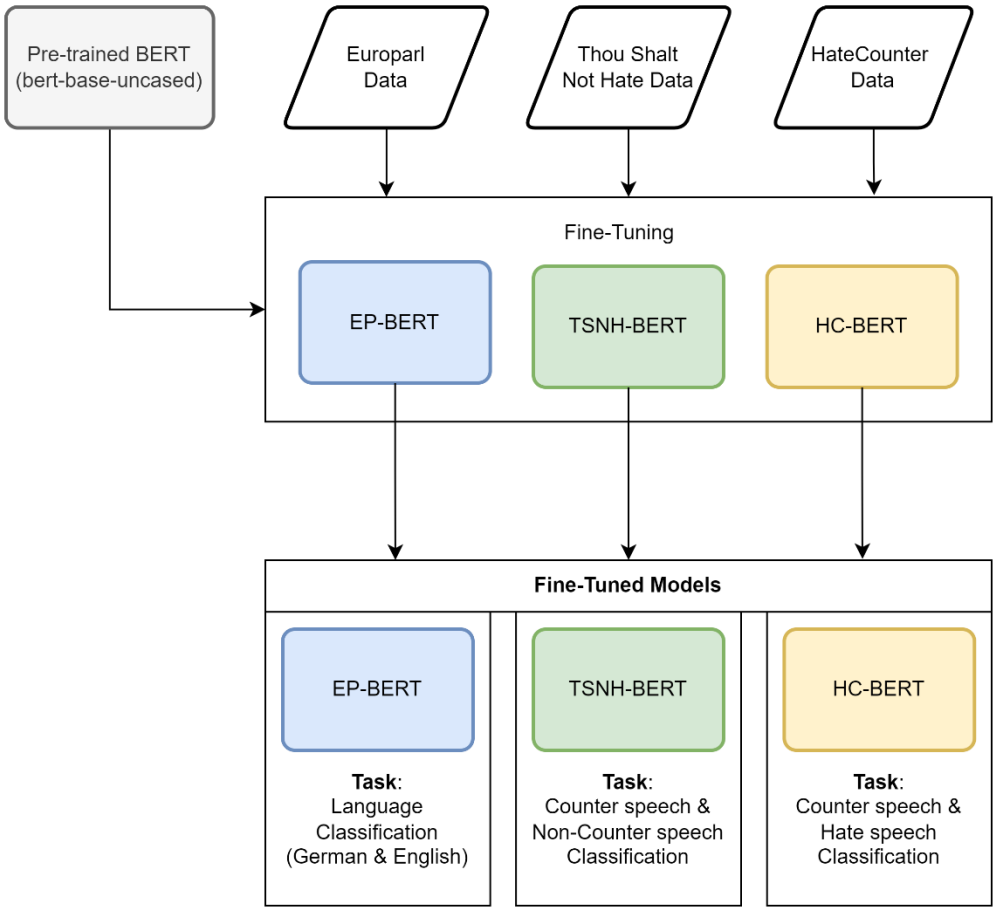


Fig. 21. Fine-tuning procedure of the three BERT classification models with their respective datasets.

The three BERT models EP-BERT, TSNH-BERT and HS-BERT got trained with a training batch size of 8 and an evaluation batch size of 16. The learning rate was set to $3e^{-5}$. Early stopping was initialized with a patience of 5. The default AdamW [151] optimizer from the Huggingface `Trainer` Class [152] was used. The three models were trained for 6 epochs before early stopping set in. The results on the performances of the models on the test set can be found in Table 10 Evaluation results of classification models

Upon discussing the training strategies of classification models EP-BERT, TSNH-BERT, and HC-BERT, the subsequent section 8.3.1 sheds light on the implementation, including an overview of the chosen hardware and software packages.

8.3.1. Implementation of Explainability Approaches

The classification and explainability approaches got implemented using the Pytorch [153] framework.
The hardware for the experimental setup consists of an Intel® Core™ i7-6900K Processor with a CPU with 3.20GHz, 128 GB RAM, and a GTX 1080 Founders Edition graphic card. Also, a 1 TB Samsung EVO SSD hard drive was part of the setup. Since the ablation study of the LIME method is computationally expensive, some of these experiments have been done on Google Colab [154] using Googles TPU.

The four explainability approaches for the experiments (LIME, Integrated Gradients, ProtoTEx and GlobEnc) have been implemented the following:

- **LIME:** The LIME package [155] was used for implementation which is based on the approach of Ribeiro et al. [156]. This package also supports visualisations of the prediction probabilities per token and their weighted importance visualised by using color gradients. Due to the high computational power that is needed for the LIME method, the number of used features for training the surrogate model was set to 10 and the number of training samples was set to 200.
- **Integrated Gradients:** Captum [128], which is an extensible interpretability library built on PyTorch [153] was chosen for the Integrated Gradients approach. For the implementation, the class `LayerIntegratedGradients` from Captum was used which provides two different ways of how the attributions for the embedding layers can be computed:
  1. Apply `LayerIntegratedGradients` and calculate the attributions in relation to the `BertEmbedding` layer of the model.
  2. Apply `LayerIntegratedGradients` for individual `word_embeddings`, `token_type_embedding` and `positional_embedding` and compute the attributions for each of these embeddings.

  Within this thesis, the first method of computing the attributions with respect to the `BertEmbedding` has been considered since this method is a higher abstraction where the individual word embeddings, token type embeddings, and positional embeddings are combined. It is used to get the overall impact of the initial input embeddings on the model's decision.
- **ProtoTEx:** The original implementation of Das et al. [12] as well as an implementation of Zhivar Sourati [18] have been adapted for training the ProtoTEx models based on BERT Transformers. Both mentioned implementations are publicly available at Github [118], [157]. The exact parameters tested in implementing this method can be found in chapter 8.4.2.
- **GlobEnc:** The original code of the paper of Modarressi et al. [11] which is available at Github [158] has been adapted.

The code of this thesis is publicy available at Github[2]. The subsequent section will delve into an evaluation of the classification model's performance and a critical analysis of the XAI methods utilized.

8.4. Evaluation Approach

The evaluation strategy in this study is designed to provide a comprehensive understanding of the XAI methods applied, incorporating both qualitative and quantitative approaches. Initially, the performance of the classification models undergoes a quantitative evaluation. In this phase, the performance of the three BERT models—TSNH-BERT, HC-BERT, and EP-BERT—that have been fine-tuned with three distinct datasets is done. Utilizing metrics such as accuracy, recall, precision and F1 score, their task-specific test sets are analysed to objectively assess their performances. Further insights into the fine-tuning approach and the datasets utilized can be found in sections 0 and 8.2 with detailed results of this quantitative evaluation outlined in section 9.1. Since ProtoTEx is not an ad-hoc method as the other chosen methods, it was pre-evaluated first. This evaluation can be found in the following section 8.4.2.
The faithfulness of the other XAI approaches is critically evaluated through an ablation study, a method inspired by similar analyses highlighted in [13]. After the quantitative analysis, a qualitative evaluation is conducted. Here, a user study is employed to delve deeper into several key aspects of the XAI methods: plausibility [20], understandability and trustfulness [21], [22], as well as the help-/usefulness [24] which are further described in section 6.3.1. This qualitative evaluation seeks to gather in-depth understanding of users' experiences and views regarding the XAI methods, presenting a comprehensive viewpoint. In-depth explanations the evaluation of the of this quantitative and qualitative method and the derived findings will be elaborated in the subsequent sections of this chapter.

8.4.1. Evaluation of Classification Performance

For the classification objective, three BERT models using three different datasets have been fine-tuned which are referred as TSNH-BERT, HC-BERT and EP-BERT within this paper. More information on the fine-tuning approach and the datasets can be found in section 8.2 and 0.
These models underwent evaluation based on their task-specific test set using the following metrics:

- **Accuracy:** This metric indicates the proportion of correctly classified instances out of the total instances.
- **Recall (Macro):** This indicates the average ability of the models to correctly identify true cases in each respective class.
- **Precision (Macro):** This denotes the average measure of the correctness of the models in classifying positive instances across different classes.

[2] https://github.com/JaquJaqu/masterthesis_XAI

- **F1-Score (Macro):** This represents the harmonic mean of macro precision and macro recall, providing a balanced view of the model's performance across different classes [159].

Detailed information on the results of the evaluation can be found in section 9.1.

8.4.2. Pre-Evaluation of ProtoTEx

In this part of the study, the evaluation procedure of the ProtoTEx model is described. This ProtoTEx method stands out from the other XAI methods since it is a prototype-based method and not a post-hoc method as the other three XAI methods. Unlike the other methods, which analyze data after the model has made its predictions (post-hoc), ProtoTEx is trained separately and creates and uses its own prototypes to help explain its own predictions. Because of this unique approach, it's necessary to assess ProtoTEx in a distinctive manner, separate from the other methods. In inference, the model considers the five nearest training samples of five determined prototypes by calculating the L2 distances in the latent space. It is essential to delve deeper into analyzing these nearest training samples to verify whether they are logical and can truly contribute to clarifying the predictions made during inference.

For this preliminary evaluation, a series of experiments are carried out using two specific datasets to assess the quality and relevance of the prototypes produced by ProtoTEx. The same pre-trained BERT model (bert-base-uncased) that was used for the classification approaches was used for training the ProtoTEx models. Along with the Thou Shalt Not Hate (TSNH) dataset, which is the counter speech related one, the Europarl (EP) dataset is considered since it includes data of a relatively easy task of detecting German and English language. The training schedules of [12] and [18] are followed for both of the models. In these settings, the number of positive and negative prototypes can be individually chosen. Regarding to the papers, 19 positive and 1 negative, and 49 positive and 1 negative prototype are recommended. As considered in the provided implementation, early stopping is used during training. One ProtoTEx model is trained on the TSNH dataset (TSNH_20t_P) for 23 epochs. Three ProtoTEx models are trained on the EP dataset (EP_50t_PT, EP_20t_PT, EP_20t_PT_2). Where EP_50t_PT trained for 50 epochs, EP_20t_PT trained also for 50 epochs and EP_20t_PT_2 trained for 169 epochs until early stopping terminated the process.

The notion of the names of the models that were trained for the experiments are as followed:

Table 9 Notation for ProtoTEx models

Notation	Description
<DATASET>…	Short name of the used dataset
<NUM_TRAIN>t	Total number of prototypes for training
<PT>	Stands for the method “ProtoTEx”
<NUM>	Serial number that is added when more than one model with the same configuration is trained (same number of prototypes for training)

<DATASET>_<NUM_TRAIN>t_<PT>_<NUM>

Different numbers of positive and negative prototypes (number of negative prototypes = number of prototypes - number of positive prototypes) have been considered during inference to observe how the number of prototypes changes the evaluation performance in different settings. Also, some of the runs were done multiple times to observe if the predicted training samples remain constant or changed when the experiments were redone. The different experimental settings are displayed in section 9.2 which is placed next to the results in Table 15 to help understanding the displayed notation of the utilized settings in the result table better.

After introducing the evaluation procedure of the ProtoTEx method we will continue with the quantitative evaluation of the remaining three XAI methods.

8.4.3. Quantitative Evaluation of XAI approaches

For the quantitative evaluation of LIME, Integrated Gradients and GlobEnc, an ablation study has been performed. The goal of the ablation study is to observe how the prediction probability of the classification model drops when words (or tokens[3]) that are considered particularly important by the respective XAI method are removed from the input. The expectation is that removing words which are considered highly important by the XAI methods, will result in a stronger change in performance (positive prediction drop) than removing less important words or tokens. By this evaluation the faithfulness of the individual methods is evaluated (see also section 6.3.1).

Preparation of the dataset

The ablation study has been done using the TSNH-BERT and the HC-BERT with their corresponding task-specific datasets. Only sentences with 10 to 30 words have been considered to ensure that the sentences consist of a sufficiently large number of words and to reduce the computational resources that are needed for the study. The prepared test set consist of 1040 sentences in total for the TSNH dataset. The HateCounter dataset consisted

[3] Note that tokens do not always have to be on a word level since the BERT tokenizer splits some unknown words into sub-word tokens. Therefore, the ablation study was conducted on token level and not on word level.

of relatively long sentences which resulted in a sample size of 57 sentences after pre-processing. Usernames and links were replaced with “<USER>” and “<LINK>”.

Ablation study
After the first iteration, which was the baseline for calculating the prediction probabilities without any token removal, the attribution scores have been calculated and the scores with their corresponding tokens were sorted in descending order. The top four positive attribution scores were then considered for the ablation study. In the following four iterations one of the four determined tokens with positive attribution scores were removed from the original text. The drop in prediction probability of the model was calculated in every iteration (after every token removal). The removed tokens were added again after calculating the prediction probability and therefore the probability drop, to make sure to just get the changes according to one removed token.

For the TSNH dataset, the attribution scores of the 1040 test sentences have been first calculated using Integrated Gradients and GlobEnc. Since LIME is computationally expensive, the data for the TSNH dataset was computed in chunks. For the HC dataset, the whole prepared test dataset consisting of 57 instances was used. Since GlobEnc can calculate attribution scores for all the 12 layers of the BERT Transformer, just the last layer was taken under consideration for calculating the scores. The steps of the attribution score calculations of the HS dataset followed the same procedure as the TSNH dataset mentioned above.

Comparison of the methods
Due to certain data instances having fewer than four positive attribution scores determined by the XAI methods, adjustments were made to ensure comparative results. The attribution scores and the predicted probabilities from each method were synchronized to include identical instances. This synchronization guaranteed that each list contained a minimum of four positive attribution scores that were used for ablation. However, this adjustment resulted in a reduced number of data instances available for comparison. Therefore, in the end a final number of 525 sentences were compared for the TSNH dataset and 38 for the HC dataset. The drop in probability and the corresponding attribution scores of the tokens/words and the Pearson correlation of the datapoint have been calculated for all the remaining test sentences. These results of the evaluation can be found in section 0.

8.4.4. Qualitative Evaluation of XAI approaches

For a qualitative evaluation, a user study was pursued by accessing the criteria of plausibility, understandability, sufficiency, trustworthiness, satisfaction, and helpfulness. The definitions of those within this work can be found in section 6.3.1. The results of the user study are displayed in chapter 9.4. The study is done using an online questionnaire consisting of two tasks:

Task 1 – Forward Simulation/Prediction:
For each of the three methods (LIME, Integrated Gradients and GlobEnc) attribution scores for five sentences are calculated. In this task, just correctly classified sentences were

considered to ease the interpretation of the explanations for the study participants. Additional added tokens (e.g. [SEP] and [CLS] tokens) which are added by the tokenizers were removed beforehand. In this setting, all the three explainability methods are visualized in the same standardized way: For each sentence, a heatmap is created by considering the calculated attribution scores. Before visualizing the attribution scores, they got normalized to -1 and 1 to ensure that the intensity of the visualized colors is the same as in the original visualizations of the methods. The colors green and violet have been chosen to show a positive (green) and negative (violet) attribution towards the models' prediction, since this color combination is not used in any of the original visualizations of the three methods and thereby avoids a visual bias to any of the methods. The more vibrant the colors are visualized, the higher the attribution is towards the prediction. An intense green color refers to a high attribution towards the predicted class and an intense violet color refers to a high negative attribution towards the predicted class. In the GlobEnc method, only the attribution scores from the last layer were considered for the explanations. Examples of all of the three methods showing the original and the standardized visualizations are shown in Fig. 22 (IG), Fig. 23 (LIME) and Fig. 24 (GlobEnc).

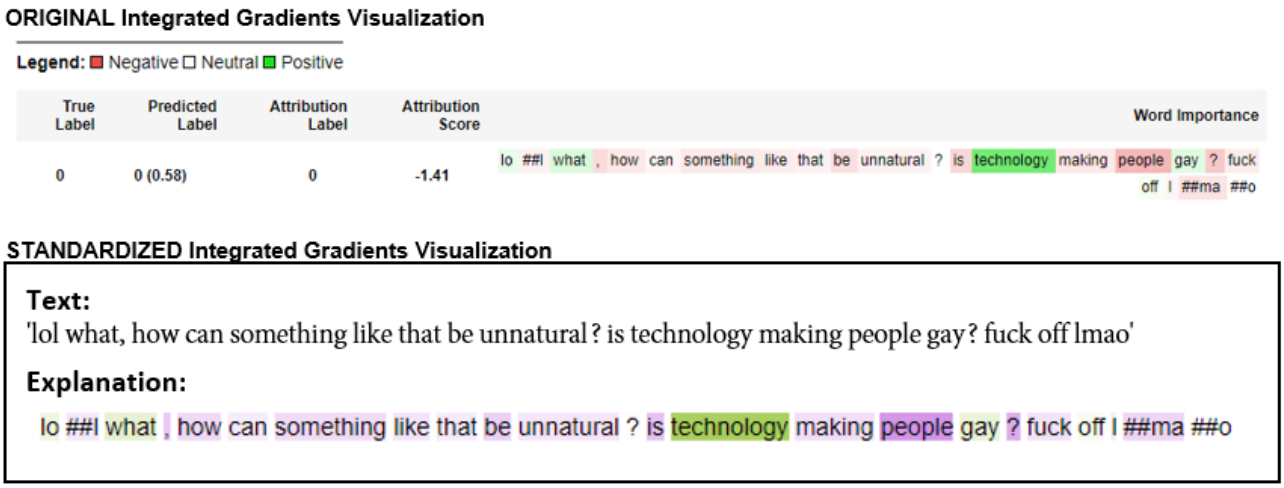
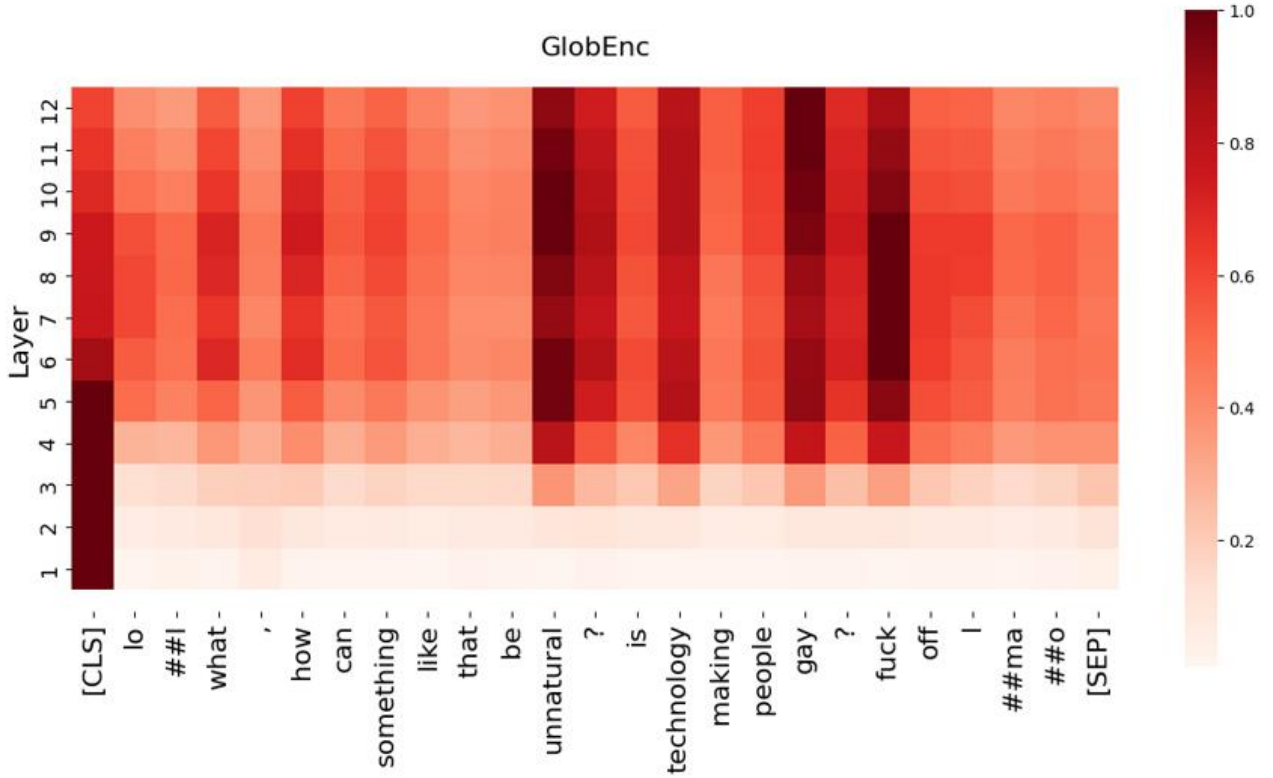


Fig. 22. Original and standardized Integrated Gradients visualization for Task 1.

Fig. 23. Original and standardized LIME visualization for Task 1.

ORIGINAL GlobEnc Visualization



STANDARDIZED GlobEnc Visualization

Text:

'lol what, how can something like that be unnatural ? is technology making people gay ? fuck off lmao'

Explanation:

lo ##l what , how can something like that be unnatural ? is technology making people gay ? fuck off l ##ma ##o

Fig. 24. Original and standardized GlobEnc visualization for Task 1.

In Task 1, the participants have to guess which prediction the model is going to make by considering the explanation/highlighted words (or tokens). In this blind study setup, the participants are not provided with the actual prediction and are not aware of the underlying methods. The texts used in the study were manually picked and the methods where randomly shown to the participants without a predefined concept. This evaluation faces the plausibility criteria. Furthermore, the participants had to answer how confident they were with their answer on a Likert scale from 1 to 5 as shown in Fig. 25.

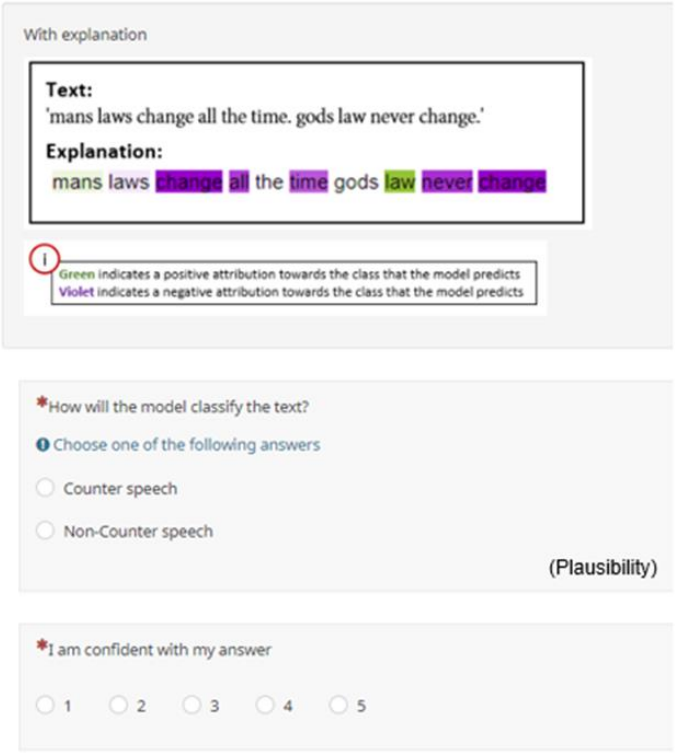
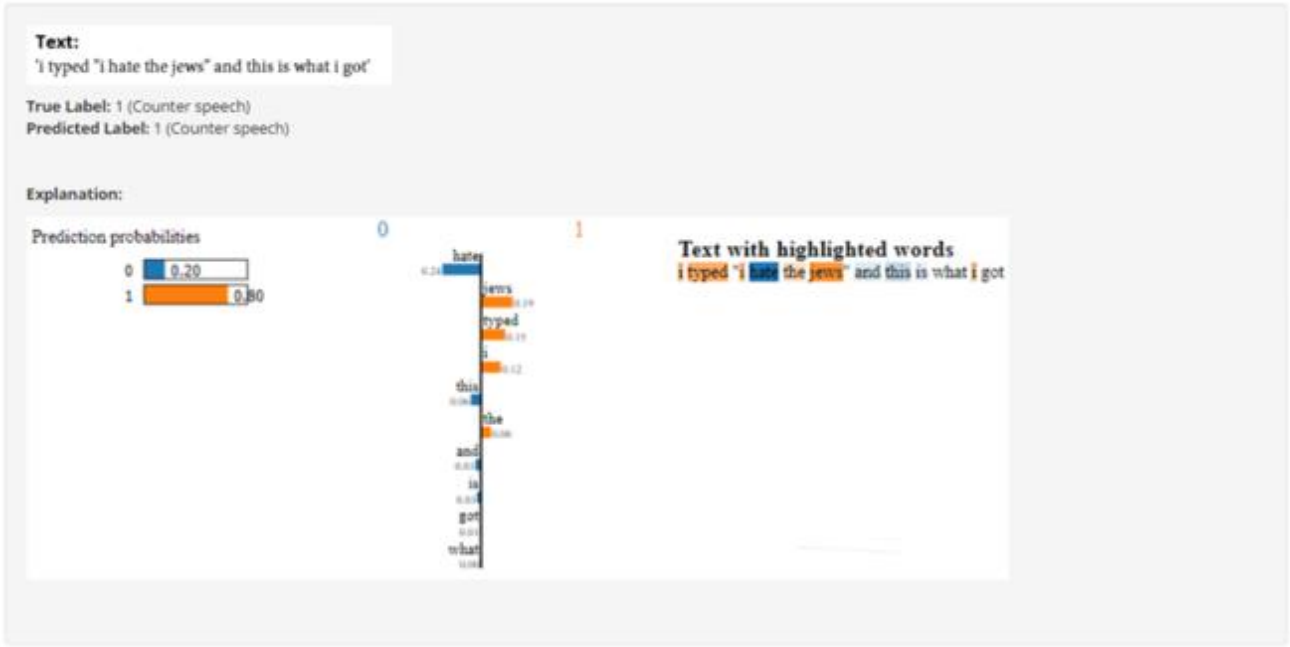


Fig. 25. Example of Task 1: Forward Prediction.

In the beginning of the task, one baseline approach for each of the five texts without any color-coding was pursued The beginning of the task started with a baseline approach where each participant were shown five texts without any color-coding. This initial phase is pivotal, allowing us to gauge the natural predictive abilities of the users without the assistance of XAI. Following this, it is possible to evaluate how much the XAI methods contribute to improving the predictive accuracy.

Task 2 – Comparative Study:

During this task of the study, participants were presented with five distinct texts (which were different ones than those from Task 1), along with the respective explanations generated by three XAI methods: IG, LIME, and GlobEnc. Also in this setting, the texts were chosen manually and the randomisation of the methods during the study was done without a concrete randomization process. To facilitate a more comprehensive evaluation, both the true label and the model's predicted label were provided to the participants. The participants were aware that they were shown different methods and that they must assess them on several criteria. The aim is to conduct a comprehensive analysis, thoroughly exploring the strengths and weaknesses inherent in each approach. To achieve this, an assessment was conducted using a Likert scale, ranging from 1 to 5, across five essential dimensions: understandability, sufficiency, trustworthiness, satisfaction, and help-/usefulness of the provided explanations. To illustrate, Fig. 26 provides a concrete example of this task, showcasing how it was executed using the LIME method.



*1.I understand why the model classified the text as 0 (Non-Counter speech)/ 1 (Counter speech) (1 = Strongly disagree, 5 = Strongly agree)

1 2 3 4 5 (Understandability)

*2. I think the visualization is sufficient for explaining why the model classified the text as 0 (Non-Counter speech)/ 1 (Counter speech) (1 = Strongly disagree, 5 = Strongly agree)

1 2 3 4 5 (Sufficiency)

*3. I think the visual explanations increases my trust in the model (1 = Strongly disagree, 5 = Strongly agree)

1 2 3 4 5 (Trustworthiness)

*4. I am satisfied with the explanation why the model classified the text as 0 (Non-Counter speech)/ 1 (Counter speech) (1 = Strongly disagree, 5 = Strongly agree)

1 2 3 4 5 (Satisfaction)

*5. I Find the explanations helpful (1 = Strongly disagree, 5 = Strongly agree)

1 2 3 4 5 (Helpfulness)

Fig. 26. Example of Task 2: Comparative Study.

After assessing the evaluation procedures of the classification models, evaluating the the ProtoTEx method as well as the ablation study and Task 1 and Task 2 of the user study to get a deeper understanding in the unique characteristics of each method, we will transition to the next chapter were the results and findings of the research are addressed.

9. Results

This chapter presents the results of the classification task, followed by a focus on ProtoTEx's preliminary evaluation which was conducted due to its unique functionality. Subsequently, we delve into the results of the ablation study and the user study, both of which were conducted for the evaluation of the XAI methods, including LIME, Integrated Gradients, and GlobEnc.

9.1. Classification

The three fine-tuned BERT models were evaluated on the test splits of their respective datasets. Also for each of the dataset a baseline was generated using the `DummyClassifier` [160] of sklearn setting the classification strategy to “uniform” which implements random baseline. The classification results of the TSNH-BERT, the HC-BERT and the EP-BERT as well as the according baselines are summarized in Table 10.

Table 10 Evaluation results of classification models

Model Name	Accuracy	Precision (macro)	Recall (macro)	F1-score (macro)
TSNH-BERT	0.7268	0.7554	0.7230	0.7166
TSNH-Baseline	0.4959	0.4958	0.4958	0.4958
HC-BERT	0.8618	0.7153	0.6360	0.6610
HC-Baseline	0.4539	0.4679	0.4353	0.3847
EP-BERT	0.9967	0.9967	0.9966	0.9967
EP-Baseline	0.5033	0.5034	0.5034	0.5033

As can be observed from the table, each of the fine-tuned BERT models performed better than their respective baselines. The TSNH-BERT model showcases better balance in performance compared to the HC-BERT model, particularly in terms of precision and recall, which are relatively close to each other. Having precision and recall values close to each other indicates that the model is proficient at accurately identifying a majority of the true positive cases while minimizing the number of false positives. In contrast, the EP-BERT model achieved near-perfect scores, exceeding 99.66%. This was expected given that the dataset was large, perfectly balanced, and purposely simplified to facilitate a straightforward solution.

Looking at the confusion matrices in Fig. 27, for the TSNH-BERT model there is a noticeable number of false positives (614) and false negatives (147), indicating the model is making

errors in both directions. While the HC-BERT model demonstrates high accuracy, it appears to be less balanced in terms of precision and recall, with a particularly lower recall. The confusion matrix indicates that the model may be biased towards predicting the positive class, and it may struggle with identifying true negatives.

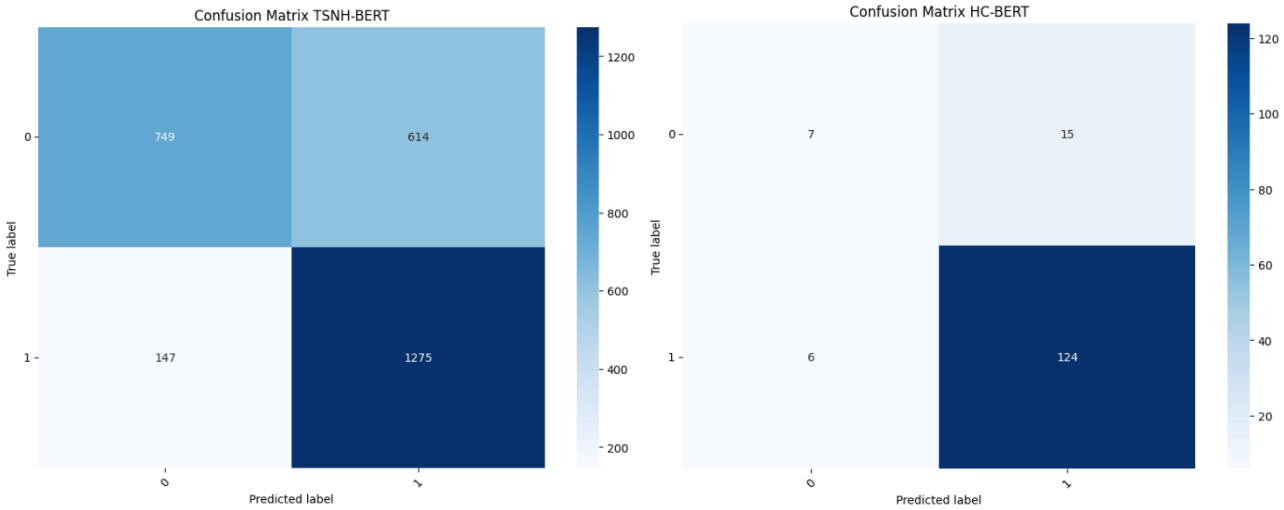


Fig. 27. Confusion matrices of fine-tuned TSNH-BERT and HC-BERT evaluated on the respective test split.

The following tables, **Table 11** Examples of misclassified indices of TSNH-BERT and HC-BERTTable 11 and Table 12 showcase correct as well as misclassified texts from TSNH-BERT and the HS-BERT.

Table 11 Examples of misclassified indices of TSNH-BERT and HC-BERT

Model Name	True Label	Predicted Label	Sentence
TSNH-BERT	1 (non-CS)	0 (CS)	Those are two disgusting swines!!! "To serve and protect." Whatever!!!
TSNH-BERT	0 (CS)	1 (non-CS)	And you don't like pussies, do you? Thought not.
TSNH-BERT	0 (CS)	1 (non-CS)	us jews are Gods chosen people. If you don't like it you are cursed. We are better than you. Americans you will need to keep dying for our prosperity. Now go kill some arabs you servant.
HC-BERT	0 (HS)	1 (CS)	the truth is: arab countries aren't a proper place to live in! no humans nor animals can live in peace here! (same thing goes with iran)

HC-BERT	0 (HS)	1 (CS)	I CANT BREATHE GOD BLESS YOU OVE RWATCH RANDOM GUY
HC-BERT	1 (CS)	0 (HS)	@user fckkk you too

Table 12 Examples of correctly classified indices of TSNH-BERT and HC-BERT

Model Name	**True Label**	**Predicted Label**	**Sentence**
TSNH-BERT	0 (CS)	0 (CS)	plesae take this down youtube.
TSNH-BERT	1 (non-CS)	1 (non-CS)	Smack that bih
TSNH-BERT	1 (non-CS)	1 (non-CS)	Damn after 4years and it popped up in my recommendation today?? ? But anyway that bitch would b e dead right now if she said al l that to me.And big ups to you brother for being cool infront of that bitch.
HC-BERT	1 (CS)	1 (CS)	They're gonna beat your ass wit h chop sticks
HC-BERT	0 (HS)	0 (HS)	They are so irritating
HC-BERT	1 (CS)	1 (CS)	How about you stick to your own country before trying to overth row a democratically elected ma yor, Arj.

As can be observed from the examples, both classification tasks pose significant challenges, primarily because the underlying meanings of the texts are often unclear, and the labels are not always intuitive. Despite these inherent difficulties, the methods achieved noteworthy performances.

9.2. ProtoTEx

In this initial evaluation, a series of experimental tests was conducted using two distinct datasets to evaluate the effectiveness and applicability of the prototypes generated by ProtoTEx. The objective was to evaluate both the quality and relevance of these prototypes, setting a foundational basis for further analysis and development. The notion of the model names of the experiments is displayed in the following table:

Table 13 Notation for ProtoTEx Experiments

Notation	Description
<DATASET>…	Short name of the used dataset
<NUM_TRAIN>t	Total number of prototypes for training
<NUM_TEST>t	Total number of prototypes for testing
<PT>	Stands for the method “ProtoTEx”
<NUM>	Serial number that is added when more than one model with the same configuration is trained (same number of prototypes for training)

<DATASET>_<NUM_TRAIN>t_<NUM_TEST>t_<PT>_<NUM>

As mentioned in section 8.4.2 multiple settings regarding the numbers of considered prototypes for training and testing are done which are displayed in the following table:

Table 14 Training and test settings of ProtoTEx models

Setting Name	Training Setting	Test Setting
SET1	n.p = 20 n.p.p = 19	n.p = 20 n.p.p = 19
SET2	n.p = 20 n.p.p = 19	n.p = 30 n.p.p = 10
SET3	n.p = 20 n.p.p = 19	n.p = 50 n.p.p = 49
SET4	n.p = 50 n.p.p = 49	n.p = 20 n.p.p = 19
SET5	n.p = 50 n.p.p = 49	n.p = 50 n.p.p = 49

*n.p. = Number of prototypes
*n.p.p. = Number of positive prototypes

The results from the various settings are shown in the in Table 15.

Table 15 Results of trained ProtoTEx models with different configurations.

Model Name	Data	Setting	Run	Accuracy	Precision	Recall	F1
TSNH_20t_20t_PT	TSNH	SET 1	1	0.4654	0.4231	0.4582	0.3915
TSNH_20t_20t_PT	TSNH	SET 1	2	0.4654	0.4231	0.4582	0.3915
TSNH_20t_30t_PT	TSNH	SET 2	1	0.5088	0.4947	0.4991	0.3717
TSNH_20t_30t_PT	**TSNH**	**SET 2**	**2**	**0.5153**	**0.5425**	**0.5225**	**0.4552**
EP_50t_20t_PT	EP	SET 4	1	0.508	0.254	0.5000	0.3369
EP_50t_20t_PT	EP	SET 4	1	0.508	0.254	0.5000	0.3369
EP_50t_50t_PT	EP	SET 5	1	0.377	0.2532	0.3828	0.2876
EP_50t_50t_PT	EP	SET 5	2	0.377	0.2532	0.3828	0.2876
EP_20t_20t_PT	**EP**	**SET 1**	**1**	**0.8707**	**0.8959**	**0.8727**	**0.8690**
EP_20t_20t_PT	EP	SET 1	2	0.8707	0.8959	0.8727	0.8690
EP_20t_20t_PT_2	**EP**	**SET 1**	**1**	**0.9803**	**0.9804**	**0.9803**	**0.9803**
EP_20t_20t_PT_2	EP	SET 1	2	0.9803	0.9804	0.9803	0.9803
EP_20t_50t_PT	EP	SET 3	1	0.508	0.254	0.5	0.3369
EP_20t_50t_PT	EP	SET 3	2	0.492	0.2460	0.5000	0.3298

* TSNH. = Thou Shalt Not Hate
* EP. = Europarl
* All metrics are macro

As can be observed from the results displayed in the tables above, the ProtoTEx models seem sensitive to the number of prototypes considered during inference, illustrating that the influence of the number of prototypes has a significant impact on the performance. Unfortunately, with different numbers of prototypes, the performance sometimes drops significantly below the random baseline of 50%, which is quite concerning given that these tasks are fundamentally 99% solvable in according to the classification results of the EP_BERT model (section 9.1). This fluctuating performance across different settings calls the methodology into question, as it seems to engender a decline in the models' capabilities, essentially causing them to degenerate. Except of the TSNH_20t_30t_PT model, the prediction results were always the same over multiple runs. This behavior of changing results was also observed in further experiments with other models when the number of prototypes varied. However, these experiments are not shown in the table and can be found in the Excel files of the ProtoTEx models provided at the Github repository[4] of this thesis. The method's goal during inference is to find sentences from the training data that are close to the predicted prototypes. Notably, it was observed that the closest training examples to the prototypes often belonged to a different class than the original test

[4] https://github.com/JaquJaqu/masterthesis_XAI

sample. This phenomenon, where ProtoTEx selects training samples from a class different than the predicted one, has also been documented in [18]. In their paper, they claimed that this behavior could occur because of the similarity of the classes in their task. However, especially in the case of a binary language classification task, this should not be the issue since distinguishing between two different languages should be an easy task and mentioned, the classification performance of the EP_BERT model was nearly perfect. This indicates that the underlying BERT Transformer should not have a problem with classifying the texts.
For a quantitative evaluation, the percentage of matches between the classes predicted for the test samples and the classes of the selected training samples representing the prototypes is calculated. This is done for the TSNH_20t_30t_PT, the EP_50t_20t_PT, EP_20t_20t_PT and the EP_20t_20t_PT_2 models, since those showed the best results on the test sets. The results are shown in Table 16.

Table 16 Matching class percentages of ProtoTEx model.

Model Name	Matching Class Percentage (True Class to Classes of Prototypes)
TSNH_20t_30t_PT	10.00%
EP_50t_20t_PT	60.87%
EP_20t_20t_PT	37.58%
EP_20t_20t_PT_2	51.85%

Unfortunately, the matching class percentages of the true label to the classes of the nearest training sample are low. The highest match can be observed in the EP_50t_20t_PT model with a percentage of 60.87%. Also, the model with the best performance EP_20t_20t_PT_2 showed this behaviour with a matching class percentage of just 51.85%. This observation raises further questions about the reliability of the selected training samples that should represent the prototypes. Unfortunately, this behavior undermines the ability to understand the models' prediction for humans, making this method less reliable and not suitable for gaining insight in the explainability of the model. Since we could not extract robust prototypes and meaningful explanations in our experiments (even not for the simple task of language classification), we excluded ProtoTEx from the further evaluation.

9.3. Ablation Study

As described is section 8.4.3, the four most important positive attribution scores according to each of the XAI methods LIME, Integrated Gradients and GlobEnc for each text were computed. Each of the determined tokens was then removed and the prediction probability of the model was calculated. At the beginning, a baseline approach where no tokens are removed was done first. Using this baseline, for every ablated token the drop in prediction probability could be calculated. The probability drop is then visualized using scatterplots to compare the attribution score of the reference token to observe how important a high or low attribution score is for the actual change in probability. This procedure was pursued for each of the two datasets, using the pre-processed test data which is described in more detail in

section 8.2. The results of the ablated sentences of the two datasets for the three XAI methods LIME, Integrated Gradients and GlobEnc are shown in the following visualizations:

TSNH-BERT (Thou Shalt Not Hate dataset):

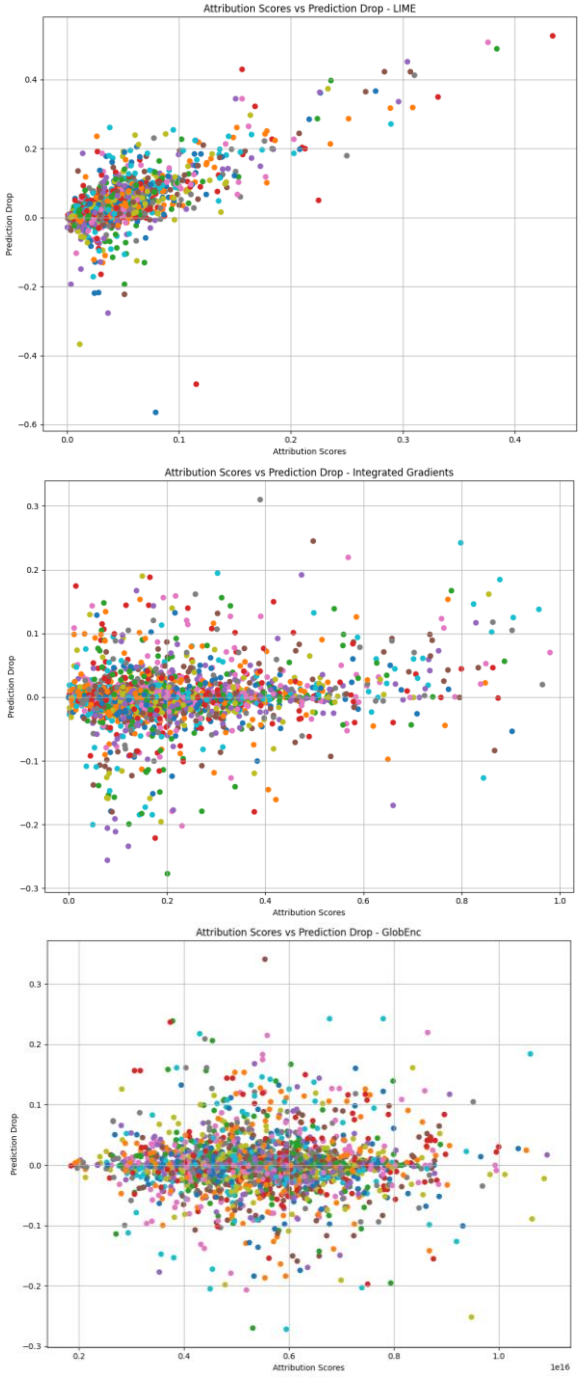


Fig. 28. TSNH-BERT: Attribution scores vs drops in prediction probability for all three XAI methods.

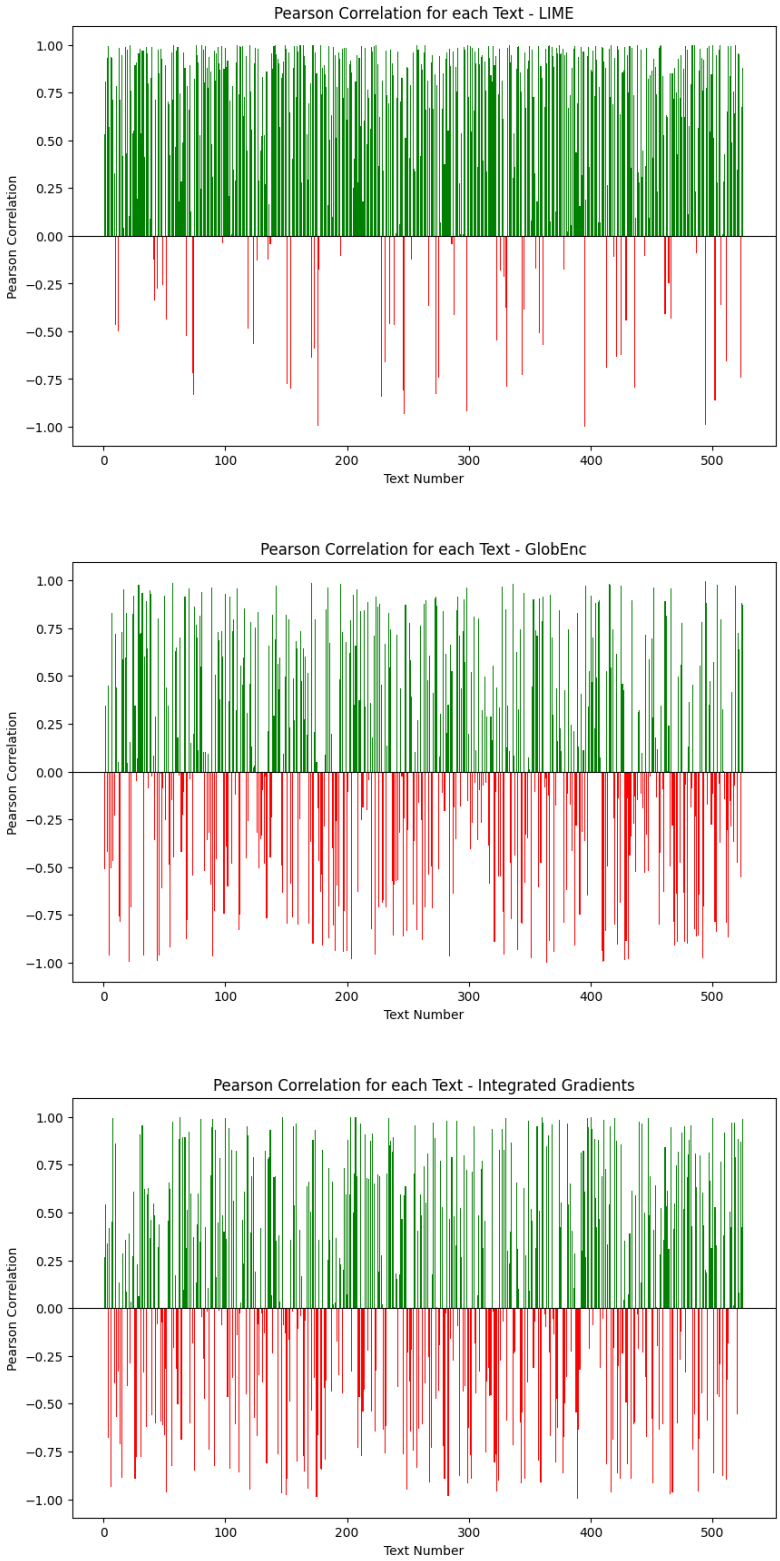


Fig. 29. TSNH-BERT: Pearson correlations for all three XAI methods.

HC-BERT (HateCounter dataset):

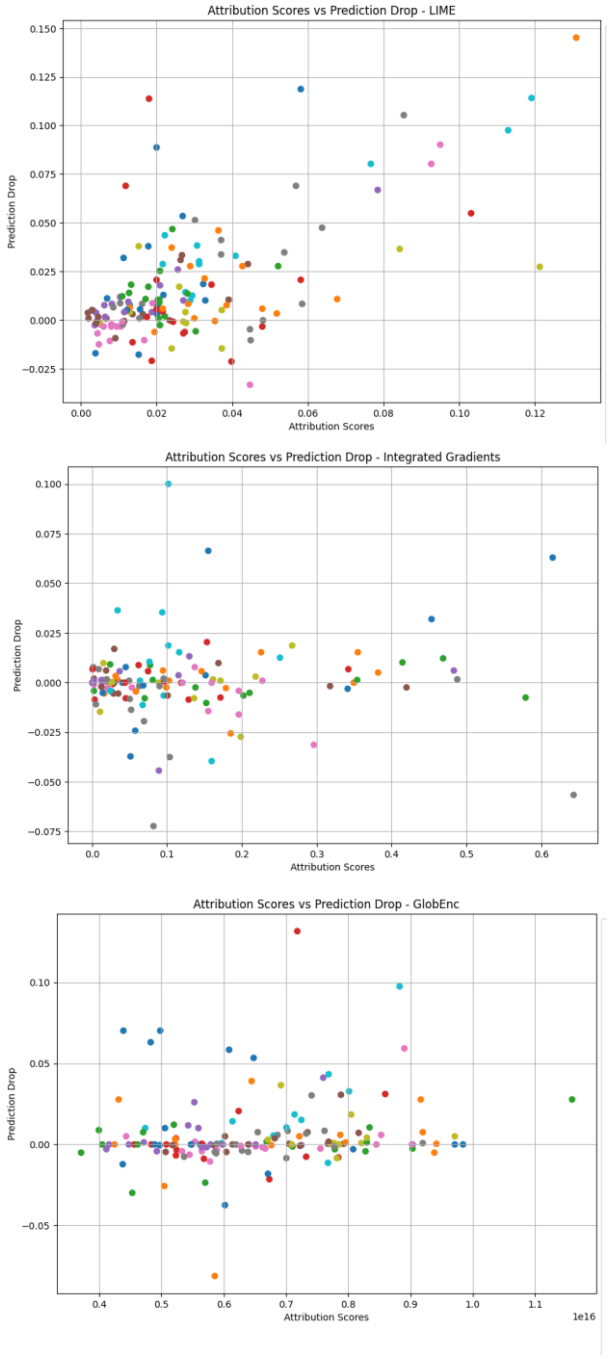


Fig. 30. HC-BERT: Attribution scores vs. drop in prediction probability for all three XAI methods.

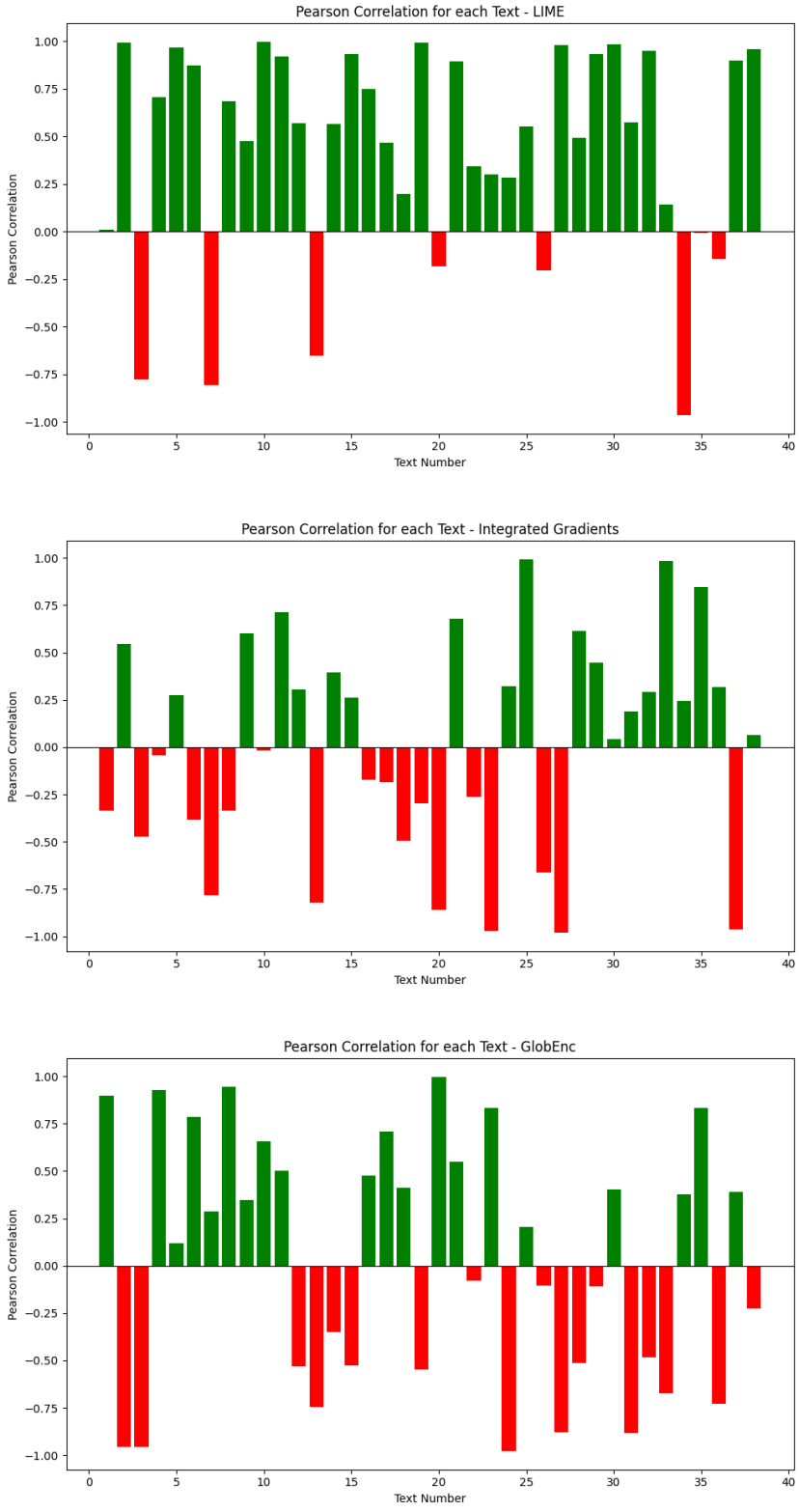


Fig. 31. HC-BERT: Pearson correlations for all three XAI methods of all three XAI methods.

A positive prediction drop indicates that the removed token has a positive impact on the prediction on the model, therefore, if the token is removed, the prediction probability of the model should decrease. However, if the probability drop appears to be a negative value,

the removed token has a negative impact on the model's decision, which indicates that the prediction probability of the model would be higher if the token would not be part of the reference sentence.

Also, the Pearson correlations are visualized to get a more in-depth insight into the results. The Pearson correlation coefficient (r) is a measure for the linear relationship of data points which is a number between -1 and 1. A positive correlation (number between 0 and 1) means as one variable changes, the other also changes in the same direction. If the correlation is negative, the changes go in the opposite directions. A number of 0 means that there is no relationship between the data points [161]. The goal of the methods would be a positive correlation towards all the data points since a removal of a token with a positive attribution should cause a prediction drop. As can be observed in Fig. 29 and Fig. 31 the IG and the GE methods show diverse results. Just the LIME method shows some correlations and therefore better results with just a few outliers with negative attributions.

Since the set goal of the ablation study was to determine the four most important positive attribution scores, which have a positive attribution towards the prediction of the model, the case of negative attributions towards the predicted label should not occur. Interestingly, all the three methods in the experiments with both models show negative probability drops and negative Pearson correlations. This indicates that some of the chosen tokens have a negative impact towards the predicted label even though their attribution score is positive. However, only tokens which positive attribution scores were removed which should decrease the performance of the model and increase the probability drop. After examining the methods with this ablation study, we move forward to the results obtained by the user study.

9.4. User study

In this section the results of the user study are shown. All additional information like the user study itself, the results as well as the code for the evaluation is provided in the GitHub repository[5] of this thesis.

For evaluating Task 1 (forward prediction), the ratio of the wrong and the right predictions of the participants are calculated. For the question regarding the confidence of the participants with their answer to Task 1 as well as for all the questions in Task 2 (comparative study) the average Likert scores per XAI method and per XAI method per text sample have been calculated to evaluate which of the score values was chosen most likely by the participants. The standard deviation of the average Likert score (also XAI method-wise and XAI method per text sample-wise) were computed. Also, the confidence intervals (CI) and the effect size (ES) (Cohen's d) have been calculated to examine the differences in the results. For a more comprehensive understanding of the obtained results, a type II ANOVA analysis was chosen since the significance of multiple factors

[5] https://github.com/JaquJaqu/masterthesis_XAI

(different XAI methods, several participants, and texts) should be analysed but not their interactions. A description of the considered metrics is given below:

Confidence Intervals:
A confidence interval gives a range around an average prediction, indicating where the true value likely lies if a study is repeated, based on a set confidence level. In this paper, the chosen confidence level is 95%. These number gives a range in which the true population mean is likely to lie. If the confidence intervals of two methods do not overlap, this indicates that there is a significant difference regarding their averaged scores [162].

Cohen's d:
Cohen's d is an effect size measure that quantifies the magnitude of difference between two group means which are standardized by the pooled standard deviation. Its value offers insights into the practical significance of an observed difference, with common benchmarks being: 0.2 (small effect), 0.5 (medium effect), and 0.8 (large effect). The magnitude of Cohen's d indicates how large the difference between the groups is. The direction of the effect size (positive or negative sign) tells which of both group has the higher mean [163].

ANOVA analysis
This analysis aims to identify how different factors affect the ratings and to determine if the observed differences are statistically significant. For that the ANOVA method is used to test the differences on two or more groups of means. The null hypothesis H_0 in the ANOVA test states that all the group means are equal. Three components (C) were evaluated:

- **C(Method):** This component was introduced to assess the effect of the three XAI methods (A (IG), B (LIME), C (GE)) on the given scores. A significant effect here would indicate that at least one method leads to different scores when compared with the others.
- **C(Text):** This aspect of the analysis looks at how the choice of text influences the scores. If significant, this would suggest that some texts were rated differently than others, regardless of the method used or the individual participant.
- **C(Participants):** This component analyses the variations in text ratings among different participants. It acknowledges that individual preferences and opinions may lead to diverse ratings, even when evaluating the same texts.

The test was implemented in python using the statsmodel ANOVA package [164] which provides the following variables per component as result:

- **Sum of Squares**: This variable represents the total variation in the data.
- **Degree of Freedom:** The number of values in the calculations that are free to vary. It's used to calculate the mean squares.

- **F-Statistic:** Is used to test if the means across different groups are equal. It calculated a ratio of the variance between the groups to the variance within the groups.
- **PR(>F) (p-Value):** Is the probability of observing an F-statistic as extreme as the one from the sample data, under the null hypothesis.
- **Small p-value** (e.g., less than 0.05) → reject the null hypothesis (there is enough evidence to believe that an effect or difference exists)
- **High p-value** (e.g., higher than 0.05) → fail to reject the null hypothesis (there is no evidence to believe that an effect or difference exists)
- **Residual**: This row reflects the within-group variation, showing the variability that is not explained by the factors that are considered in the analysis.

By examining these components, the analysis helps to identify the separate effects of the method, text, and participant on the ratings. This offers a clearer understanding of what is causing the differences we see in the evaluations [165], [166], [167]. For the ANOVA test within this paper a significance level alpha of 0.05 (or 5%) has been chosen.

9.4.1. Participants

In the user study, 14 persons contributed to the qualitative evaluation in this thesis. Most of the participants had full-time positions, were students or part-time workers. Of these, 8 possessed master's degrees, 4 had completed further education or held bachelor's degrees, and 2 were advanced graduates or held PhDs. All the 14 participants had some understanding of AI. Additionally, all were familiar with XAI concepts. The age distribution ranged from 18 to 44, with the majority falling between 25 to 35 years. In terms of gender, there were 4 females, 9 males, and 1 non-binary person. None of the participants exhibited colour blindness, which was critical for being able to contribute to the user study.

9.4.2. Task 1 – Forward Simulation/Prediction:

In Task 1 where the participants were asked to guess what the model will predict based on the given explanation, it can be observed from Table 17, Table 18 and Fig. 32 that each of the explanations lead to the participants to make a right prediction more often, compared to the baseline without explanation.

Table 17 Percentage of correctly predicted classes by the participants of Task 1 of the user study

Right Predicted	Baseline	A (IG)	B (LIME)	C (GE)
Total Score	27	38	36	36
Percentage	38,57%	54,29%	51,43%	51,43%

Table 18 Percentage of wrong predicted classes by the participants of Task 1 of the user study

Wrong Predicted	Baseline	A (IG)	B (LIME)	C (GE)
Total Score	43	32	34	34
Percentage	61,43%	45,71%	48,57%	48,57%

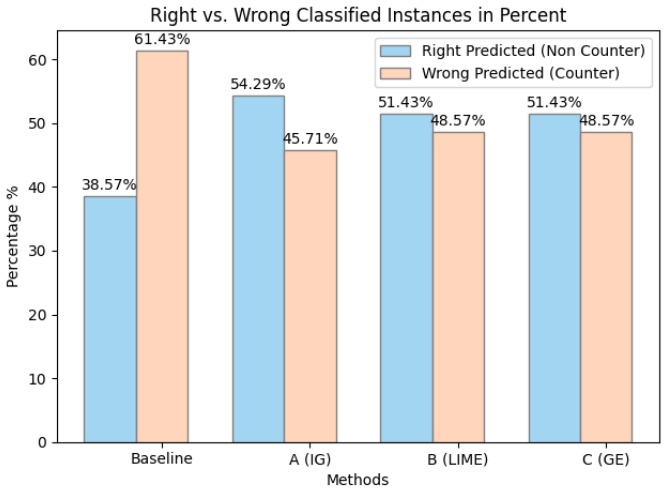


Fig. 32. Right vs wrong classified instances of Task 1 of the user study.

Without the help of the explanations only 38,57% of the samples where correctly classified (Baseline). However, the right forward prediction using IG was 54,29%, using LIME was 51,43% and GlobEnc also helped the participants to achieve a percentage of 51,43% of correctly predicted samples.

The following graphs illustrate the calculations done for the additional question about the confidence of the participants giving their answer in the evaluation.

Additional Question - Confidence:
Question: *“I am confident with my answer.” (1 = Strongly disagree, 5 = Strongly agree)”*

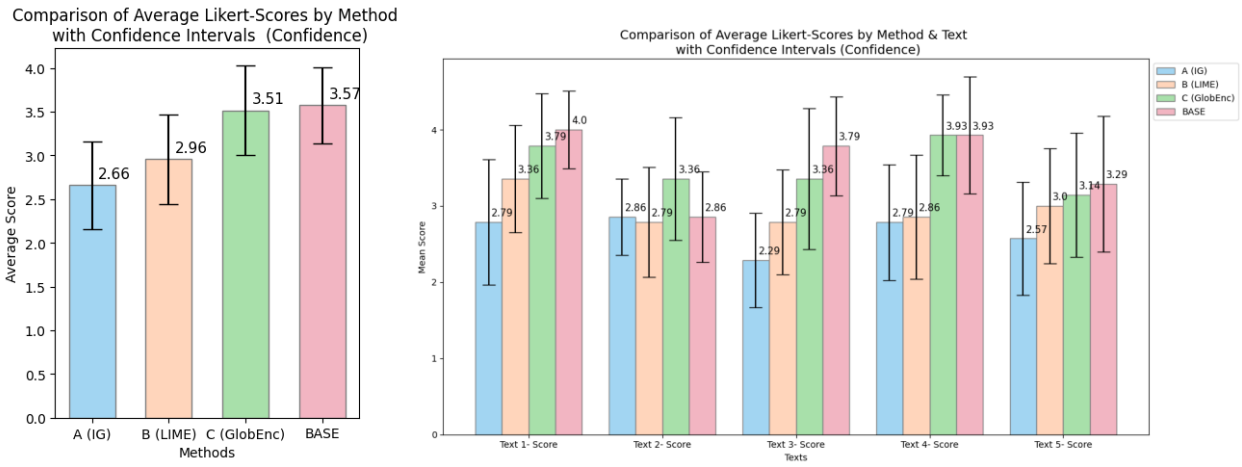


Fig. 33. Comparison of the average Likert scores for the confidence of the participants during Task 1.

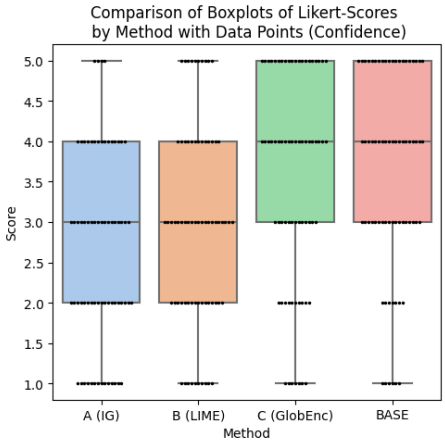


Fig. 34. Comparison of boxplots showing the average Likert scores for the confidence of the participants during Task 1.

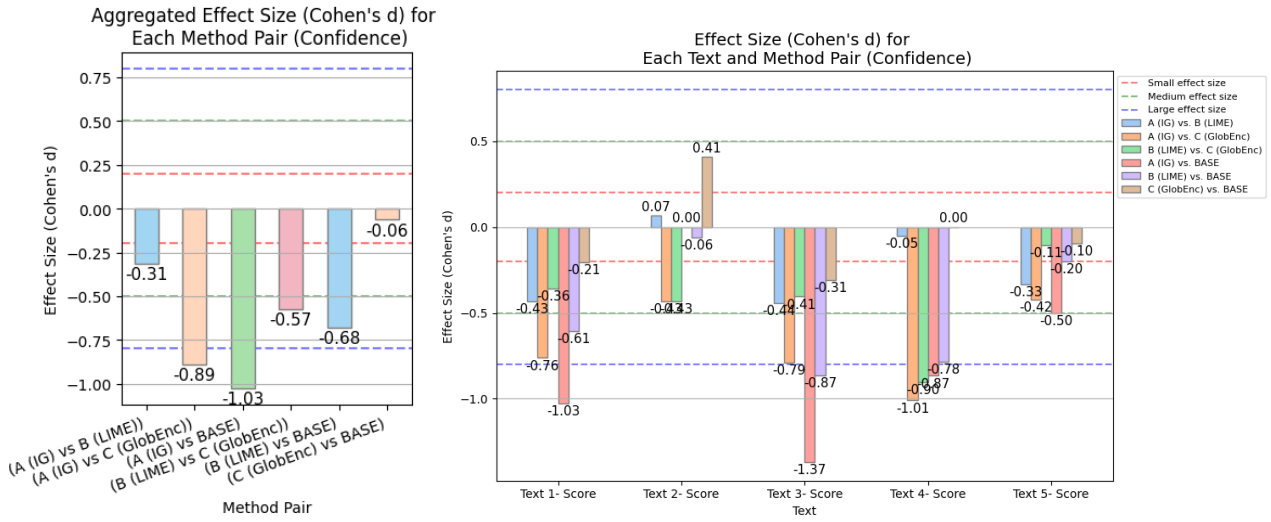


Fig. 35. Comparison of the effect size (Cohen's d) for the confidence of the participants while answering Task 1.

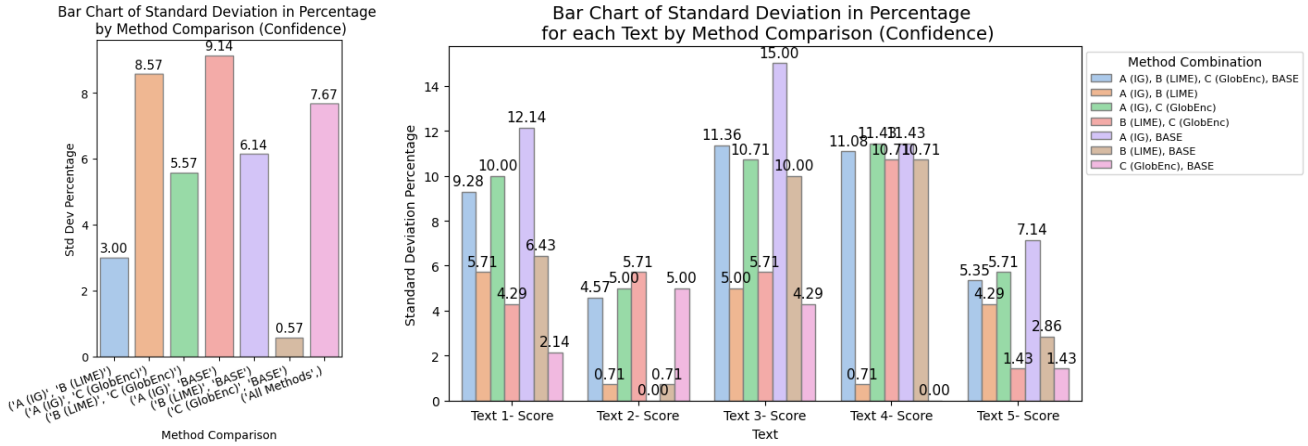


Fig. 36. Comparison of the standard deviation of the average Likert scores of the confidence of the participants while answering Task 1 of the user study.

Table 19 ANOVA - Test of Task 1 of the user study (Confidence).

	Sum of Squares	Degree of Freedom (df)	F-Statistic	PR(>F) (p-Value)
C(Method)	41.15	3.0	14.24	$1.288416e^{-08}$
C(Text)	12.55	4.0	3.26	$1.252486e^{-02}$
C(Participants)	171.18	13.0	13.67	$5.251104e^{-23}$
Residual	249.55	259.0	NaN	NaN

Average Likert score:
Since the Likert scale is ordinal by nature, not the percentage but the average Likert scores are shown in the visualizations. These are visualized to get a better understanding in which score was chosen more likely by the participants. Interestingly, the results show that the participants were more confidence in their forward prediction without any additional explanations since the average Likert score of the baseline (BASE) reached the highest score. The GlobEnc method was given most likely a higher score in providing confidence compared to LIME and Integrated Gradients. The median and quartiles from IG and LIME were similar. Also, the results of GlobEnc to the baseline showed similar outcomes.

Standard deviation (SD) of average Likert score in % and effect size (ES):
It is noteworthy that in terms of the standard deviation and the effect size of the average Likert score during model-wise comparison, the baseline (BASE) and GlobEnc (GE) showed minimal differences. Nevertheless, the text-wise comparison showed some inconsistency across the five texts. The highest effects were shown in the comparison of IG vs GE (ES = -0.89) and IG and BASE (ES = -1.03).

ANOVA-Test:
The ANOVA analysis in Table 19 ANOVA - Test of Task 1 of the user study (Confidence).shows that the method, text and the participants have all a statistically significant effect on the dependent variable, which are the raw inputs of the participants. This conclusion is made because of the low p-values of each of the methods which are below the set 5% alpha threshold.

9.4.3. Task 2 – Comparative Study

As described in chapter 8.4.4, in Task 2, the three original representations of the explainability methods were analysed. The participants of the user study were asked for their opinions to gain insight into how understandable, sufficient, trustworthy, satisfactory, and help-/useful the explanations are on a Likert scale ranging from 1 (= strongly disagree) to 5 (= strongly agree).
The computed graphs according to the results of this analysis regarding the mentioned criteria are displayed and the highlights shown are commented in this chapter.

Question 1 - Understandability:

Question: *"I understand why the model classified the text as 0 (Non-Counter speech)/ 1 (Counter speech) (1 = Strongly disagree, 5 = Strongly agree)"*

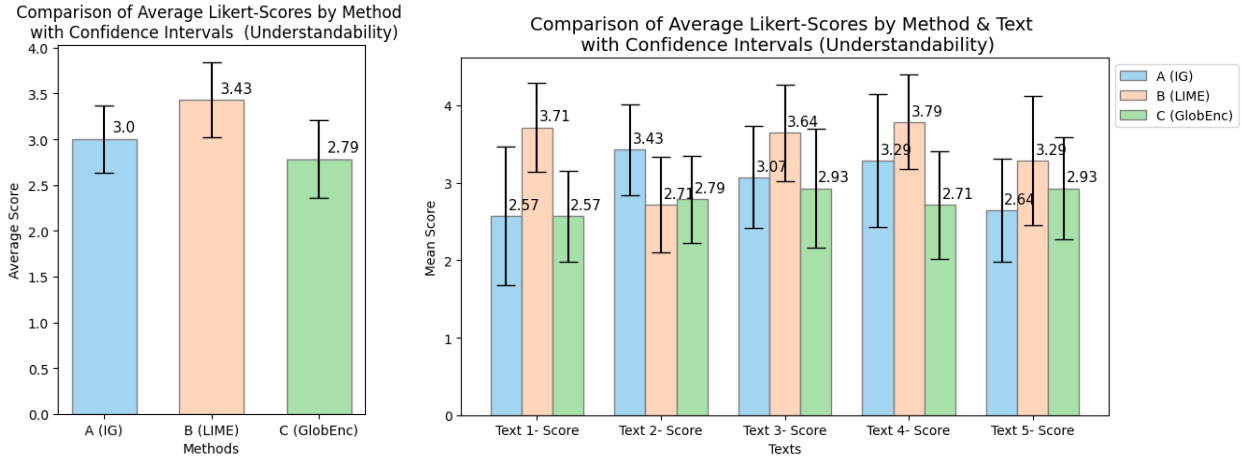


Fig. 37. Comparison of the average Likert scores for the understandability criteria of the XAI methods.

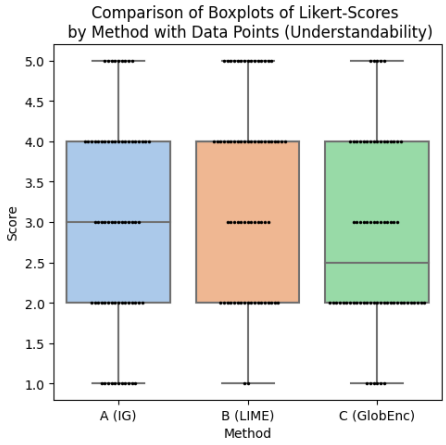


Fig. 38. Comparison of boxplots showing the average Likert scores for the understandability criteria of the XAI methods.

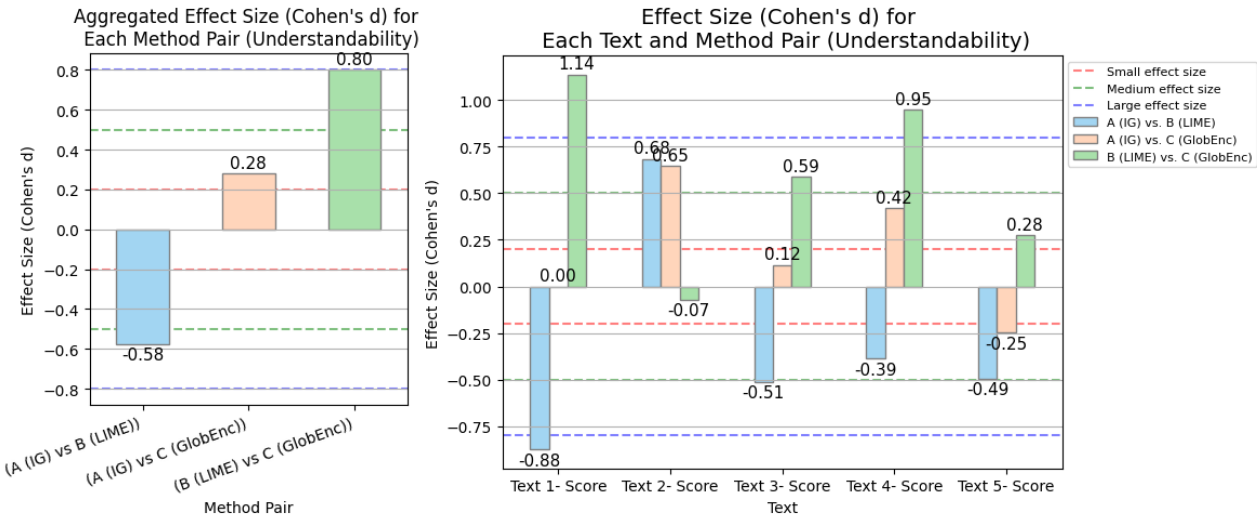


Fig. 39. Comparison of the effect size (Cohen's d) for the understandability criteria of the XAI methods.

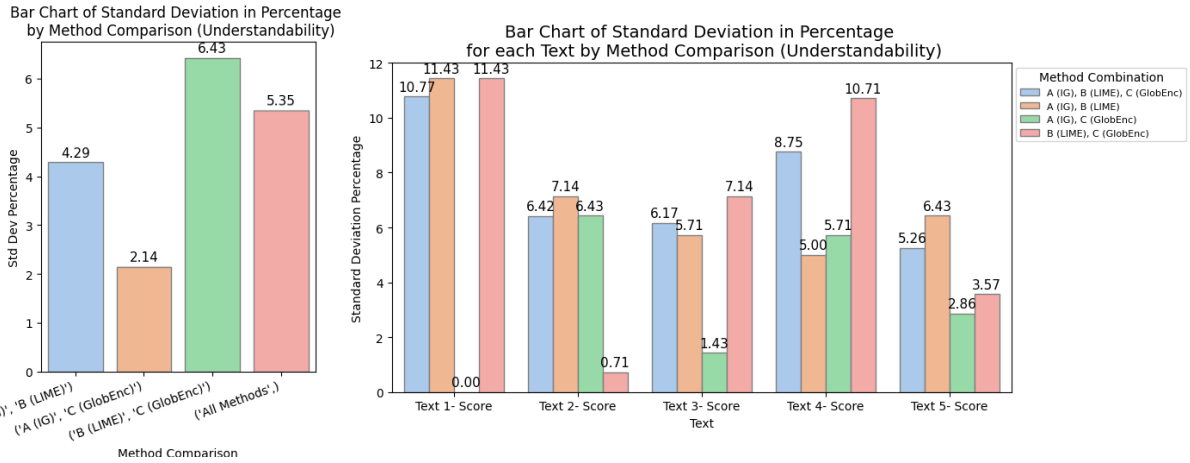


Fig. 40 Comparison of the standard deviation of the average Likert scores for the understandability criteria of the XAI methods.

Table 20 ANOVA - Test of Task 2 of the user study (Understandability)

	Sum of Squares	Degree of Freedom (df)	F-Statistic	PR(>F) (p-Value)
C(Method)	15.00	2.0	6.57	$1.742858e^{-03}$
C(Text)	3.95	4.0	0.87	$4.857063e^{-01}$
C(Participants)	76.07	13.0	5.12	$8.913304e^{-08}$
Residual	216.91	190.0	NaN	NaN

Average Likert score:
The average Likert score of the three methods was the highest for LIME with a Likert score 3.43, followed by IG with 3 and GE with 2.79.

Standard deviation (SD) of average Likert score in % and effect size (ES):
In the pairwise comparisons the results of the approach comparing the IG and the GE method showed the least differences with a total percentage of 3% in SD. However, LIME and IG showed in comparison the highest difference of 6%.

ANOVA-Test:
The ANOVA analysis highlights that differences in the methods used in the study contribute to statistically significant variations in ratings ($p = 1.742858e^{-03}$, $F = 6.57$). This confirms that the choice of method matters to the ratings of the participants. In contrast, the text factor was found to have no significant effect on ratings ($p = 0.4857$, $F = 0.87$). Importantly, individual differences among participants were found to be a strong contributing factor to the variations in ratings ($p = 8.913304e^{-08}$, $F = 5.12$). This may emphasize the role of personal preferences and biases in the given scores.

Question 2 - Sufficiency:

Question: *“I think the visualization is sufficient for explaining why the model classified the text as 0 (Non-Counter speech)/ 1 (Counter speech) (1 = Strongly disagree, 5 = Strongly agree)”*

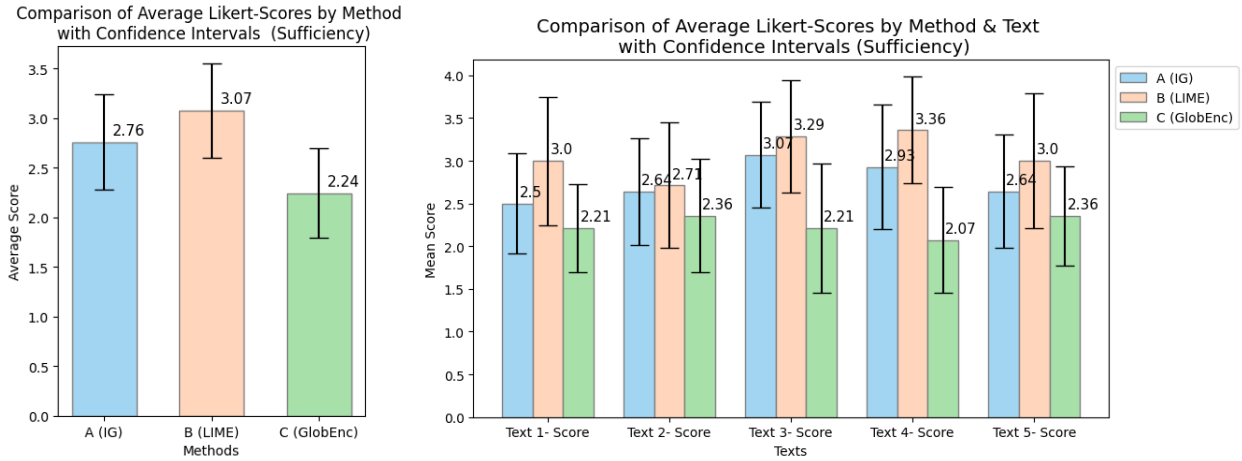


Fig. 41 Comparison of the average Likert scores for the sufficiency criteria of the XAI methods.

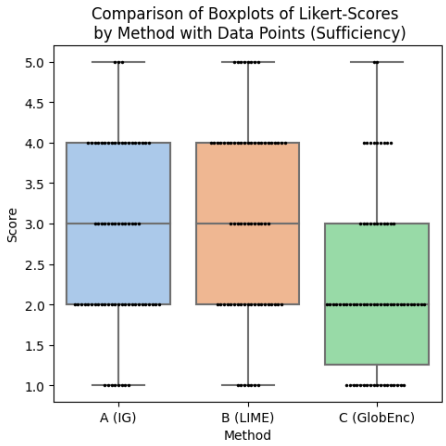


Fig. 42. Comparison of boxplots showing the average Likert scores for the sufficiency criteria of the XAI methods.

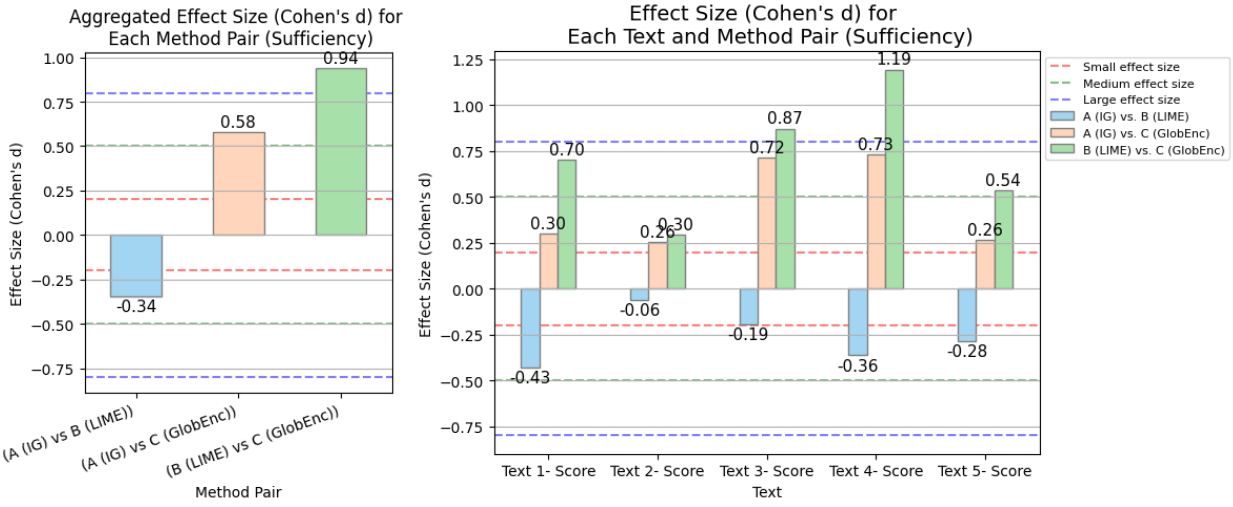


Fig. 43. Comparison of the effect size (Cohen's d) for the sufficiency criteria of the XAI methods.

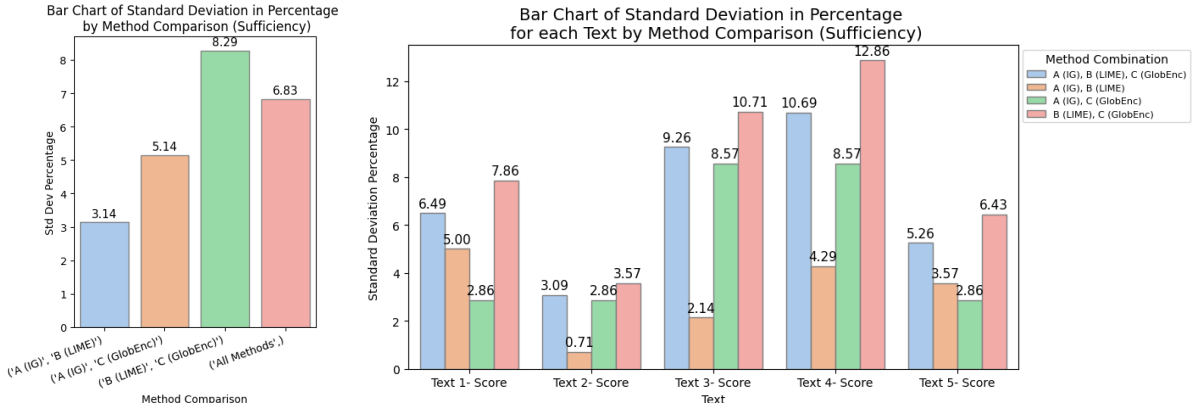


Fig. 44 Comparison of the standard deviation of the average Likert scores for the sufficiency criteria of the XAI methods.

Table 21 ANOVA - Test of Task 2 of the user study (Sufficiency)

	Sum of Squares	Degree of Freedom (df)	F-Statistic	PR(>F) (p-Value)
C(Method)	24.50	2.0	13.35	$3.766463e^{-06}$
C(Text)	2.76	4.0	0.75	$5.574899e^{-01}$
C(Participants)	89.28	13.0	7.48	$7.983725e^{-12}$
Residual	174.34	190.0	NaN	NaN

Average Likert score:
For evaluating the sufficiency of the methods, the average Likert score of the three methods was the highest in the LIME model with 3.07, followed by IG with 2.76 and GE with 2.24. By observing the average Likert scores for each text and method a trend which is followed by each of the texts can be observed.

Standard deviation (SD) of average Likert score in % and effect size (ES):
The pairwise comparison of LIME and GE showed the most differences with a standard derivation of 8% and an effect size of 0.94.

ANOVA-Test:
The ANOVA analysis revealed significant variations in ratings regarding the methods used in the study ($p = 3.766463e^{-06}$, $F = 13.35$), confirming the influence of different methods on the observed ratings. However, differences in the texts used did not result in statistically significant variations in the given scores ($p = 0.5575$, $F = 0.75$). Additionally, a strong significance was found among the factor of the participants ($p = 7.983725e^{-12}$, $F = 7.48$), illustrating the important role of individual differences in rating behavior among the participants

Question 3 - Trustworthiness:

Question: *“I think the visual explanations increases my trust in the model (1 = Strongly disagree, 5 = Strongly agree)”*

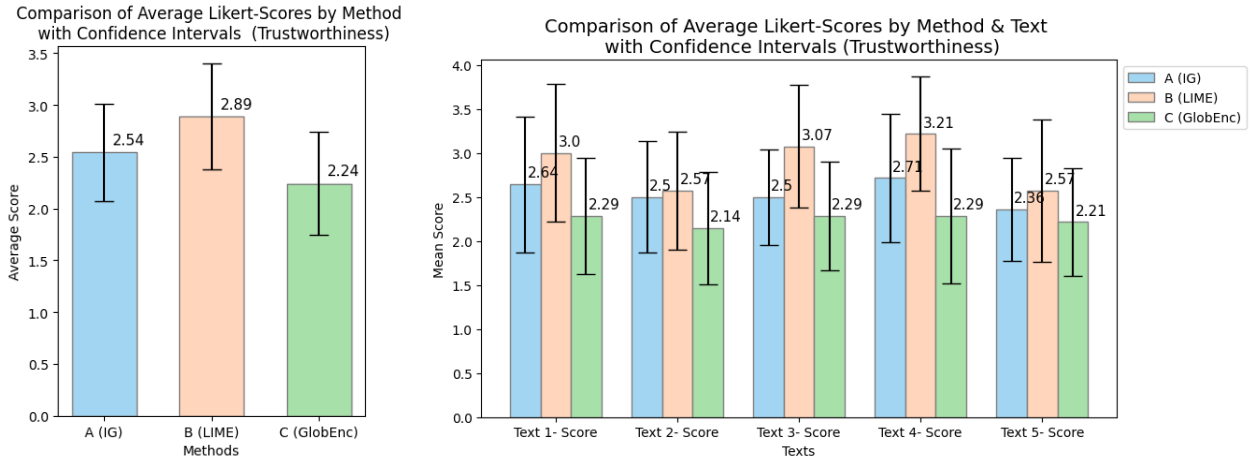


Fig. 45. Comparison of the average Likert scores for the trustworthiness criteria of the XAI methods.

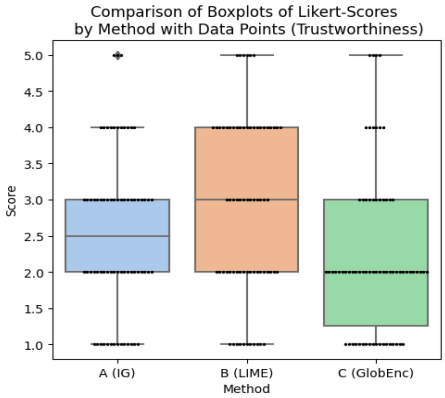


Fig. 46. Comparison of boxplots showing the average Likert scores for the trustworthiness criteria of the XAI methods.

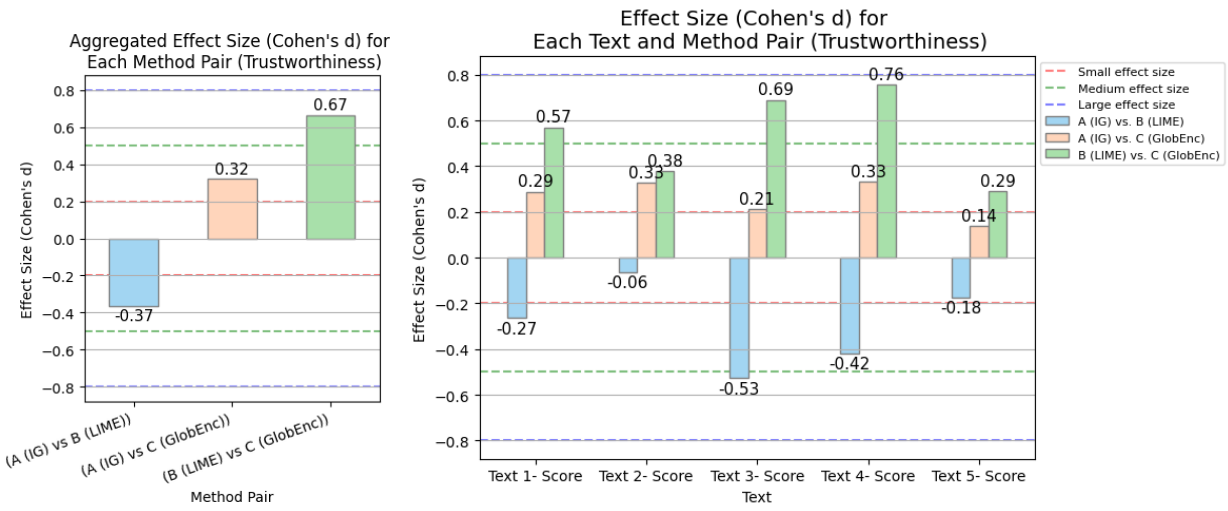


Fig. 47. Comparison of the effect size (Cohen's d) for the trustworthiness criteria of the XAI methods.

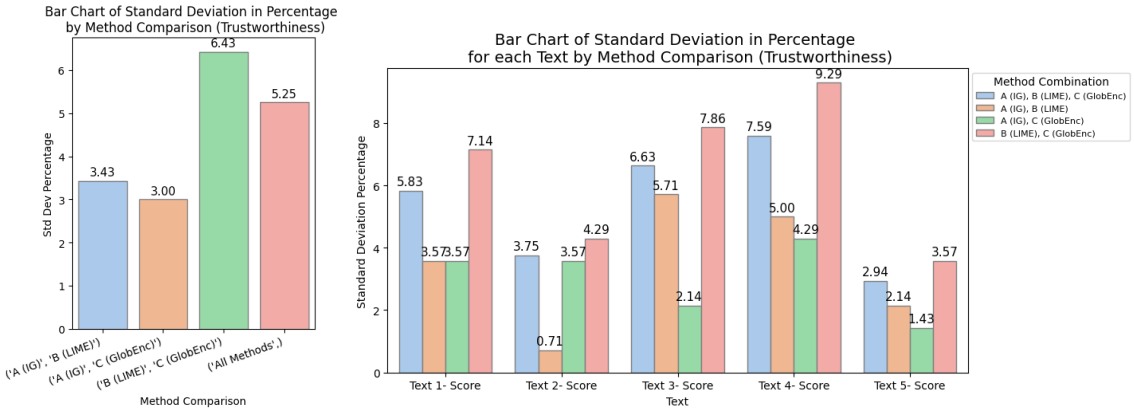


Fig. 48. Comparison of the standard deviation of the average Likert scores for the trustworthiness criteria of the XAI methods.

Table 22 ANOVA - Test of Task 2 of the user study (Trustworthiness)

	Sum of Squares	Degree of Freedom (df)	F-Statistic	PR(>F) (p-Value)
C(Method)	14.49	2.0	9.45	$1.225463e^{-04}$
C(Text)	4.12	4.0	1.34	$2.548708e^{-01}$
C(Participants)	127.55	13.0	12.80	$5.727276e^{-20}$
Residual	145.66	190.0	NaN	NaN

Average Likert score:
A similar trend as in the evaluation of the sufficiency can be observed here. The average Likert score of the three methods was the highest in the LIME model with 2.89, followed by IG with 2.54 and GE with 2.24. By observing the average Likert scores for each text and method a trend which is followed by each of the texts can be observed.

Standard deviation (SD) of average Likert score in % and effect size (ES):
LIME and GE showed the most differences with a standard derivation of 6% in the pairwise comparison of these both and an effect size of 0.67.

ANOVA-Test:
The results of the ANOVA analysis highlighted significant differences in the ratings between the methods used ($p = 1.225463e^{-04}$, $F = 9.45$), which shows the influence of the methodological differences on the given scores. In contrast, the specific texts utilized in the study were found not to affect the ratings in a statistically significant manner ($p = 0.2549$, $F = 1.34$). The strong significance in the participants' factor ($p = 5.727276e^{-20}$, $F = 12.80$) shows individual differences in the rating behavior among the participants.

Question 4 - Satisfaction:
Question: *"I am satisfied with the explanation why the model classified the text as 0 (Non-Counter speech)/ 1 (Counter speech) (1 = Strongly disagree, 5 = Strongly agree)"*

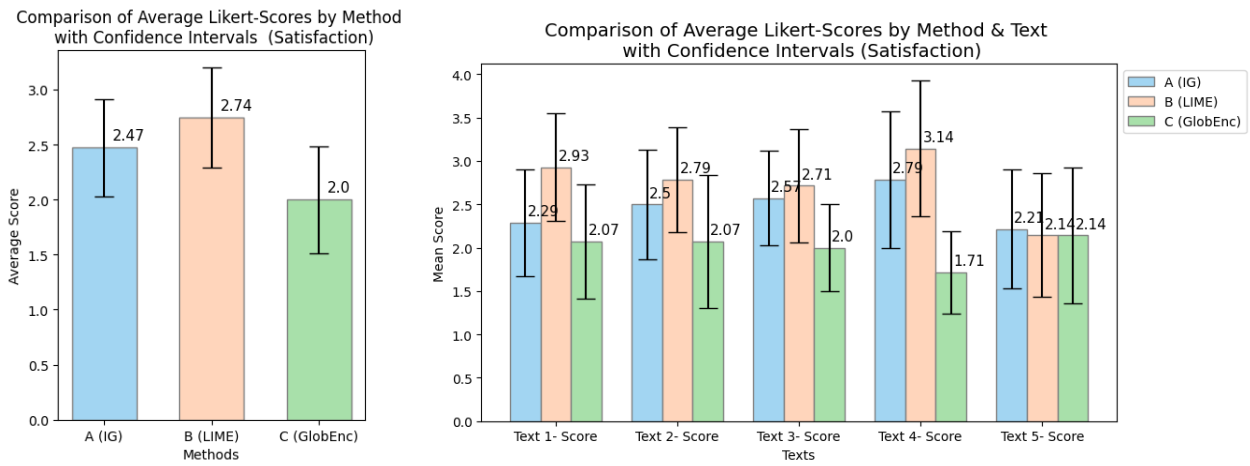


Fig. 49. Comparison of the average Likert cores for the satisfaction criteria of the XAI methods.

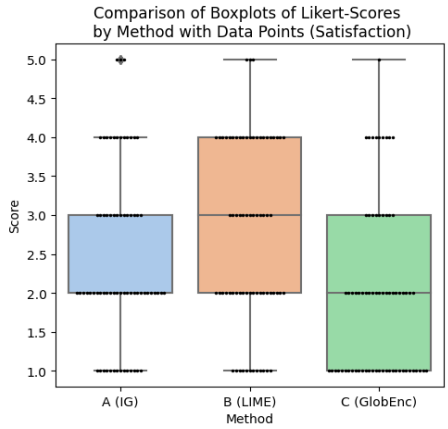


Fig. 50. Comparison of boxplots showing the average Likert scores for the Satisfaction criteria of the XAI methods.

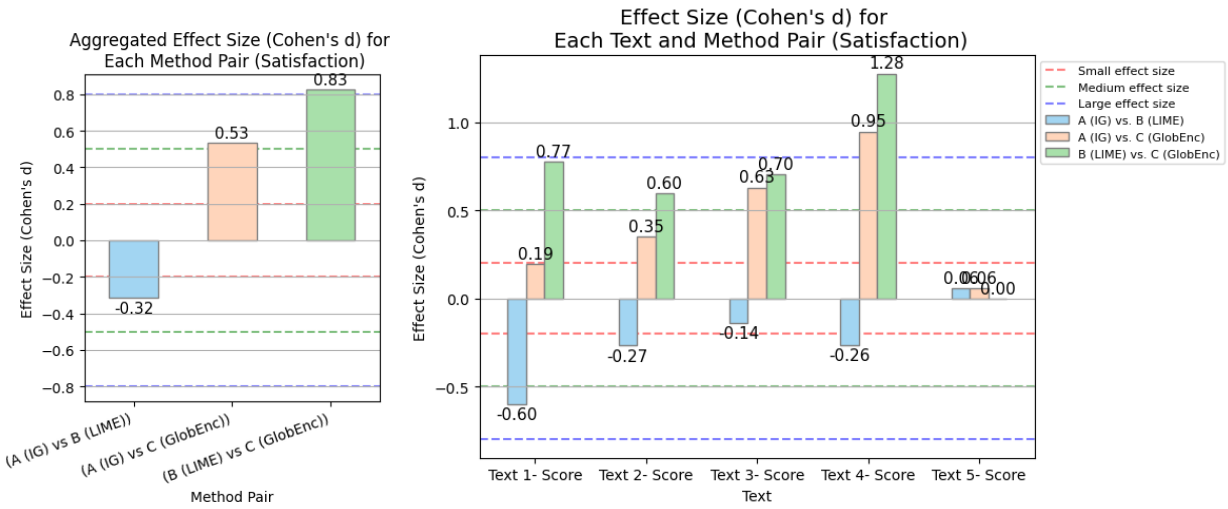


Fig. 51. Comparison of the effect size (Cohen's d) for the satisfaction criteria of the XAI methods.

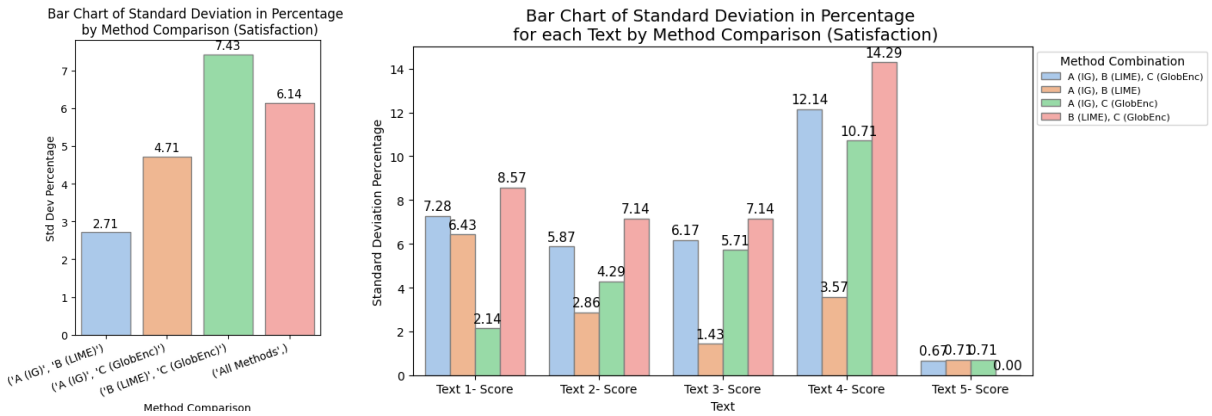


Fig. 52 Comparison of the standard deviation of the average Likert scores for the satisfaction criteria of the XAI methods.

Table 23 ANOVA - Test of Task 2 of the user study (Understandability)

	Sum of Squares	Degree of Freedom (df)	F-Statistic	PR(>F) (p-Value)
C(Method)	19.78	2.0	10.29	$5.699372e^{-05}$
C(Text)	3.38	4.0	0.88	$4.771778e^{-01}$
C(Participants)	82.86	13.0	6.63	$2.161911e^{-10}$
Residual	182.57	190.0	NaN	NaN

Average Likert score:
The average Likert score of the three methods was the highest in the LIME model with 2.79, followed by IG with 2.47 and GE with 2.00.

Standard deviation (SD) of average Likert score in % and effect size (ES):
LIME and GE showed the most differences with a standard derivation of 7% in the pairwise comparison of these both (ES = 0.83), followed by a 5% deviation of the IG compared to the GE method (ES = 0.53).

ANOVA-Test:
The ANOVA analysis revealed that the method used significantly affected the ratings (p = 5.699372e-05, F = 10.29), indicating differences in the ratings between different methods. However, the text itself did not show a significant impact on the ratings (p = 0.4772, F = 0.88), suggesting that the specific texts used in the study did not lead to variations in ratings. Moreover, the analysis indicated a significant difference among participants (p = 2.161911e-10, F = 6.63), reflecting individual variations in rating behavior.

Question 5 – Help-/Usefulness:

Question: *“I find the explanation helpful (1 = Strongly disagree, 5 = Strongly agree)”*

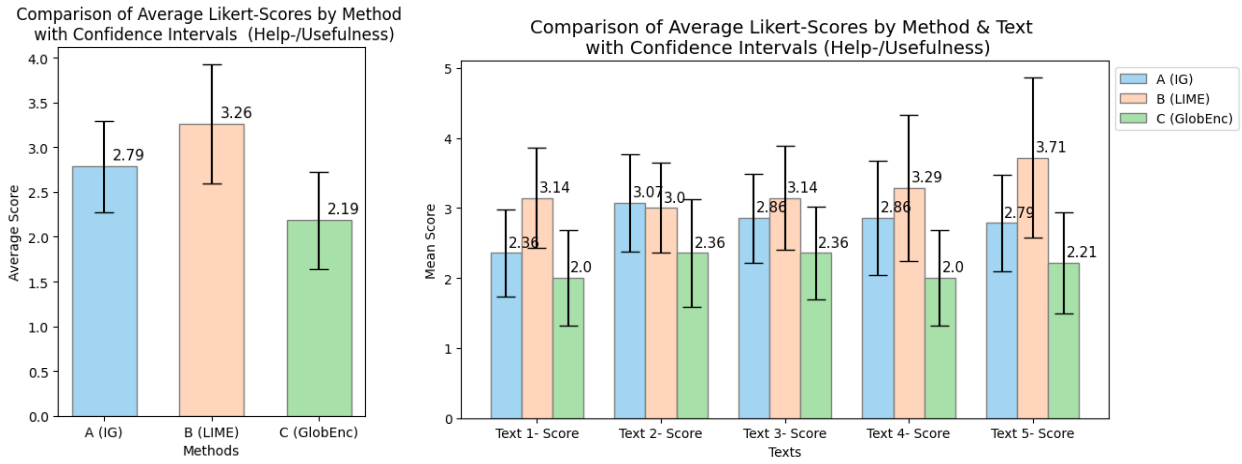


Fig. 53 Comparison of the average Likert scores for the help-/usefulness criteria of the XAI methods.

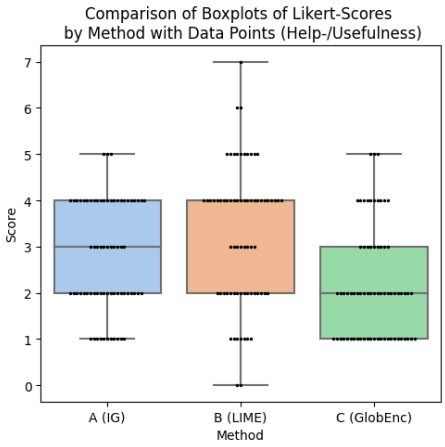


Fig. 54. Comparison of boxplots showing the average Likert scores for the help-/usefulness criteria of the XAI methods.

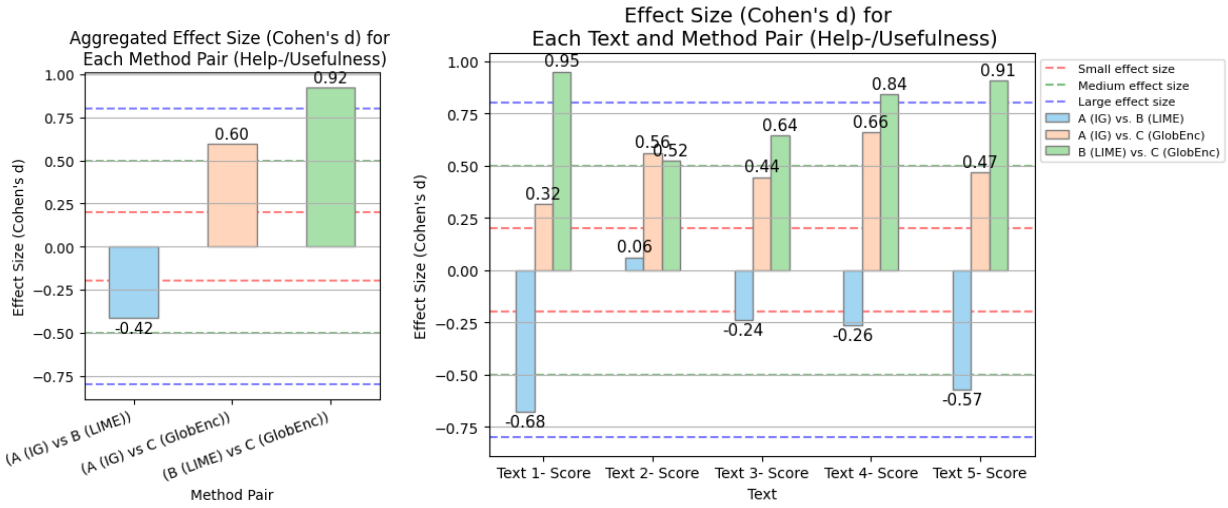


Fig. 55. Comparison of the effect size (Cohen's d) for the help-/usefulness criteria of the XAI methods.

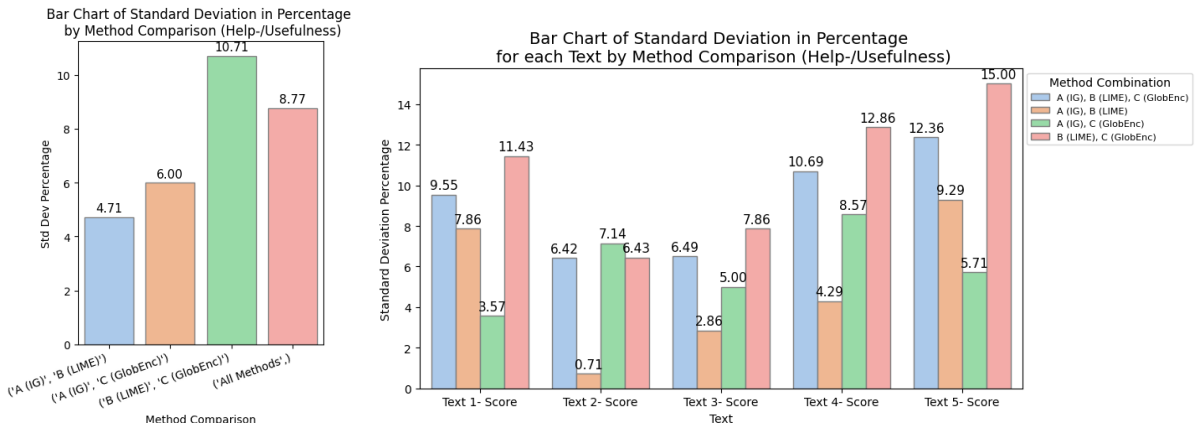


Fig. 56. Comparison of the standard deviation of the average Likert scores for the help-/usefulness criteria of the XAI methods.

Table 24 ANOVA - Test of Task 2 of the user study (Helpfulness)

	Sum of Squares	Degree of Freedom (df)	F-Statistic	PR(>F) (p-Value)
C(Method)	40.37	2.0	18.50	$4.575318e^{-08}$
C(Text)	3.88	4.0	0.89	$4.722049e^{-01}$
C(Participants)	140.51	13.0	9.90	$1.056391e^{-15}$
Residual	207.35	190.0	NaN	NaN

Average Likert score:
Also here, the average Likert score of the three methods is the highest in the LIME model with 3.26, followed by IG with 2.79 and GE with 2.19.

Standard deviation (SD) of average Likert score in % and effect size (ES):
LIME and GE also showed again the most differences with a standard derivation of 10.71% in the pairwise comparison (ES = 0.92), followed by a 6% deviation of the IG compared to the GE method (ES = 0.49).

ANOVA-Test:
The ANOVA analysis revealed a statistically significant effect of the chosen method on the ratings ($p < 0.0001$, $F = 18.50$), indicating that different methods led to different chosen scores. However, the effect of different texts on the ratings was not significant ($p = 0.4722$, $F = 0.89$), suggesting that the specific text did not influence the ratings. Additionally, there was again a significant effect of participants on the scores ($p < 0.0001$, $F = 9.90$).

The goal of this section is to display the results of the classification approaches, the ProtoTEx pre-evaluation, the ablation study as well as the user study. In the evaluation of the user study, the aim of the textual descriptions of the different metrics are to highlight the most important observation shown in the graphs. A detailed discussion of the observation made in the evaluation of the classification tasks as well as the evaluation regarding the XAI methods follows in the next chapter.

9.5. Discussion

After detailing the methodology, outlining the evaluation strategy, and presenting the results, this section delves into dissecting and interpreting the observed outcomes. The observations regarding the fine-tuned classification models and the conducted experiments, the evaluation of the ProtoTEx method, the ablation study as well as the results of the user study are described in more detail.

9.5.1. Classification

This section continues with discussing the results of the classification approaches of the fine-tuned BERT models. Upon analysing the results, it was shown that all the fine-tuned models outperformed their task specific random baseline as displayed in Table 25.

Table 25 Evaluation results of classification models

Model Name	Accuracy	Precision (macro)	Recall (macro)	F1-score (macro)
TSNH-BERT	0.7268	0.7554	0.7230	0.7166
TSNH-Baseline	0.4959	0.4958	0.4958	0.4958
HC-BERT	0.8618	0.7153	0.6360	0.6610
HC-Baseline	0.4539	0.4679	0.4353	0.3847
EP-BERT	0.9967	0.9967	0.9966	0.9967
EP-Baseline	0.5033	0.5034	0.5034	0.5033

Also, it was evident that the fine-tuned TSNH-BERT model outperformed other counter-/hate speech related models in key performance metrics like precision, recall, and F1-score when evaluated on its respective dataset. However, despite these advantages, its accuracy was slightly inferior to that of the HC-BERT model. The TSNH-model had achieved an accuracy of 72.68%. It is assumed that the accuracy of 86.18% in the HC-BERT model is higher in comparison with the other metrics because of the high-class imbalances in the dataset. Furthermore, the high classification results of the EP-BERT with an accuracy of 99.67% were expected since the task of language classification is considered as easy, and the used dataset was relatively large and perfectly balanced. The performances were considered as good enough to use the models for evaluating the XAI methods which discussion follows in the next chapter.

9.5.2. XAI Methods

ProtoTEx

The evaluation of the XAI methods, especially for the ProtoTEx model was experimental as described in detail in section 8.4.2. This method was from special interest since it provides explanations during inference by providing trained sentences from the trained clusters as

explanation. The method is also considered as faithful due to its case-based reasoning strategy.
Several experiments with different model settings where done. Especially the EP dataset was considered more intensively for the evaluation because of its high evaluation performances in the trained EP-BERT model, and since distinguishing between two languages is a simpler task than classifying e.g., counter and non-counter speech. So, any errors regarding the classification behavior of the model which would affect the performance of the explanations should be better prevented. As mentioned, the ProtoTEx method aims to identify sentenced from the training data that align closely with the predicted prototypes. However, the determined prototypes kept belonging to different classes (the opposite class in binary classification) compared to the original test sample. As noted in [18], a similar behavior was observed where they found overlapping prototypes related to a specific label. They mentioned that this might occur due to the similarities between the classes of their approach. Even though the language detection task was introduced to prevent this problem, it did not show any significant benefit. Since the matching class percentages (as shown in Table 16 Matching class percentages of ProtoTEx) of the evaluated models were too low, the ProtoTEx method was excluded from any further evaluation and is considered as not beneficial for the understanding of the models behavior for humans. Another finding was that, since it is not a post-hoc method and lacks dedicated libraries for its implementation, the time needed for its implementation and training can be time-consuming, depending on the complexity of the underlying model. Nevertheless, previous research has shown that the classification accuracy remains comparable to the underlying models. This implies that the generation of explanations has little to no impact on the model's performance [12], [18]. This observation could also be pursued in some, but not all the experiments within this thesis. However, the findings of this research showed that the performance of the model is highly depended on the total number of prototypes and the considered amount of positive and negative prototypes.

The other three XAI methods Integrated Gradients, LIME and GlobEnc were evaluated by an ablation study and user study. For that, attribution scores of the tokens of the texts where calculated using the methods. These should in theory indicate which tokens have a positive or negative attribution towards the predicted label of the model. This approach is discussed below.

Ablation study
The ablation study was introduced to evaluate the faithfulness criteria of the methods. In this case, a method can be considered as faithful when the removal of a token has an influence on the prediction depending on the value of the attribution score. In theory, a high attribution score should lead to a prediction drop when the respective token is removed from the sentence.

The results displayed in section 8.4.3 which were evaluated using TSNH-BERT and the HC-BERT show, that all the methods had negative probability drops even though all the chosen tokens should have a positive impact towards the model prediction. Therefore, a removal of those should increase the probability drop in any case. The Integrated Gradients and the GlobEnc methods both tend to select tokens which have little or no impact on the

predicted probability of the model. In contrast, the LIME method shows slightly better results since the ablated datapoints showed some correlation in the scatter plot. This indicates that tokens with higher attribution scores cause indeed a higher drop in the probability of the models when removed. This means that the chosen tokens are relevant for the decision of the model. However, the results of each of the methods are relatively poor. Especially Integrated Gradients and GlobEnc show no significant benefit on the importance of the chosen tokens based on the attribution scores. The findings indicate that all the methods should not be considered as faithful, even though LIME showed the best results in comparison to Integrated Gradients and GlobEnc.

User study

The user study was considered to evaluate the criteria of plausibility, understandability, sufficiency, trustworthiness, satisfaction and help-/usefulness. Two tasks were introduced in this assessment: Task 1, the forward prediction task and Task 2, the comparative study.

Task 1 was the forward prediction task which faces the plausibility criteria. In this task, the participants were shown the sample texts with and without standardized explanations (the tokens were highlighted in the same way regarding the attribution scores of the methods).The results displayed in chapter 9.4 indicate that all of the provided explanations had indeed a benefit in helping the participants to choose the right prediction, even though the results show that their confidence decreased when explanations were provided. In comparison, the GlobEnc explanation gave them a slightly higher level of confidence compared to the Integrated Gradients and the LIME method. However, since the attribution scores indeed helped the participants to make right forward predictions, all the methods can be considered as plausible.

The results of Task 2, which was the comparative study shows that for each of the five remaining criteria, LIME reached the highest chosen average Likert scores, followed by Integrated Gradients and afterwards the GlobEnc method. During pairwise comparisons of the methods, the highest standard deviation of the average Likert score per method were shown in the LIME and the GE method ranging of 6-8% which shows that the participants' choices varied more in these methods. A lower standard deviation would suggest they often chose the same value in different settings.

Also, the ANOVA test showed in all cases that the chosen methods have a significant influence on the given Likert scores. The texts did not cause statistically significant variations. This indicates that the text are valid candidates for the evaluation. The factors among the participants where strongly significant which confirms different rating behaviors among the participants. Since the Likert scale had a length of 1 (= Strongly disagree) to 5 (= Strongly agree), a positive influence on the specific criteria is given when the average Likert score of the models in greater than 3. However, only the criteria of understandability and sufficiency reached higher scores than 3. The results of the questions regarding the other criteria were below this threshold. For the understandability criteria, LIME reached an average Likert score of 3.42 followed by Integrated Gradients with 3.00. For the sufficiency criteria, only the LIME method reached a Likert score close but still slightly higher than the mid which is a score of 3.07. The other criteria were rated below 3 with the lowest score of

2 of GlobEnc regarding the satisfaction criteria. It is noticeable that LIME performed best and the GlobEnc method performed the worst in all 5 cases.
These results show that regarding to the set threshold of 3, only two of the criteria are slightly fulfilled. Those are the understandability and the sufficiency criteria by the LIME method. However, the chosen confidence intervals as well as the visualized boxplots of the average Likert scores show that the results of the methods are not clearly distinguishable. Consequently, the findings indicate certain tendencies rather than definitive outcomes.

These results conclude that the standardized explanation of Task 1 had a benefit for making better decision towards the forward prediction task. However, when the participants were shown the original visualizations of the methods only the LIME method has a certain benefit for the participants. One of the reasons for this result could be that the participants have used the LIME method before and are more experienced using it since it is a common technique. However, this assumption is not measured within this thesis.
The task of counter- and/or hate speech detection is a difficult one for humans as well as for ML models. Some of the participants reached out afterwards and described what their problems were during the study. Some of them claimed that they find it hard to make choices during the user study since they missed a context for the provided text and had some issue understanding the exact intention of the shown text. Therefore, even though the results of Task 2 are poor, it can't be said that the explainability methods are the primary concern.
However, except of the LIME methods which showed a little benefit in making the model more explainable and interpretable, the others, ProtoTEx, Integrated Gradients and GlobEnc are not suitable for the task of hate-/counter speech detection.

10. Conclusio and Future Work

The study investigated in an extensive literature research in the methods of XAI in BERT classifiers in the use case of counter- and/or hate speech detection in text, as well as in a comparison of four different XAI methods that can be used to explain such models.
The research questions to be answered are:

- *Which explainability methods can be used to explain the decisions of Transformer-based language models?*
- *How do different explainability concepts differ and what are the strengths and weaknesses?*
- *Which methods are most helpful in result interpretation for humans?*

To answer the first question, the variety of attempts to explain the decisions of a Transformer-based model are diverse. The most popular methods which have been studied for Transformer models are gradient-based, perturbation-based, attention-based and the use of counterfactuals. But also, the use of prototypes has found some but little interest in research in the past years. Among the most classical methods are for sure LIME and SHAP, but also variants of those are getting more popular (e.g., DeepLiftSHAP, KernelSHAP), Attention-based methods have been from special interest in the past few years since the attention-mechanism is an inherently part of the Transformer architecture, making it a seemingly good candidate for the use for interpretability and explainability purposes.

To answer the second question, the concepts, and functionalities of the different methods of explainability techniques are not always clearly distinguishable since some of the concepts can be found in other methods. In perturbation-based methods the input features get altered and the shift in the models' prediction gets observed. The more the prediction changes, the more influence it has on the prediction. These methods, especially the often-used method LIME is computationally expensive. The advantage of LIME is that it is a well explored method which can be easily adapted using several open-source libraries. Attention-based methods use the power of the built- in attention mechanism by considering the attention weights for determining which features the model finds crucial for a decision. Even though these techniques are considered as rather fast, the literature showed that the attention mechanism may not necessarily highlight the exact reasoning behind a model's decision. The often lack of directional information and can sometimes be misled by the context rather than the actual content, making their explanations not always trustworthy. Gradient-based methods show more promising results regarding Transformer models. In these methods, the gradients from input to output features get explored throughout the model. However, the application to Transformers, as demonstrated in the literature review, is not without its challenges, particularly in terms of faithfulness and consistency. Prototypes, on the other hand, seek to provide explanations by identifying instances from the training data that closely resemble the given input, hoping to give a more intuitive understanding through examples. Even though this method is considered

faithful by design, it is found that current methods often lack in being useful for human interpretability since the determined prototypes do not always have to be from the same class as the predicted class of the model. Also, since there are no libraries for this prototype-based approach, the implementation could be time consuming and difficult depending on the underlying model structure. There is clearly no one-fits all solution for all models and use cases.

To answer the last question, four methods of the mentioned categories have been chosen and further explored. The chosen methods were Integrated Gradients, LIME, GlobEnc and ProtoTEx. However, after some initial experiments with ProtoTEx, this method had to be discarded from further evaluation since the chosen prototypes kept being from other classes than the target or predicted class. This behavior made it non-beneficial for this comparison. The results of the evaluation of the remaining methods showed, that even though all the three methods can be considered as plausible in the use case of counter speech detection, none of the methods gave sufficient results regarding the five criteria of understandability, sufficiency, trustworthiness, satisfaction, and use-/helpfulness. However, the LIME methods showed at least some tendencies for being beneficial regarding the understandability and the sufficiency criteria.

The result show that the explainability in Transformer models in such complex tasks as hate- and counter speech detection has still plenty of room for improvement. Since the complexity of Transformer models is complicated and gets even more complex in newer models, the development of sophisticated tools to interpret and visualize the model components, architecture, and underlying data is needed. Even though there are already some tools available, more investigation should be made in evaluating the benefits of this methods. Given the observed strengths and weaknesses of current methods, future studies should consider combining multiple explainability techniques to potentially harness their collective strengths and offset individual limitations. Since gradient-based methods seemed promising in past literature, more research in combinations of gradient- and attention-based approach should be explored. The combination of different model architectures could also be beneficial for more consistence and more faithful explanations. Although the ProtoTEx method was not included in subsequent evaluations, it warrants further investigation. Future research should emphasize determining prototypes exclusively from the target class to gain a clearer comprehension of the provided prototypes. This could give users a more intuitive understanding of the chosen prototypes. Given the feedback of the user study, it is evident that interpretability is also a challenge for user experience. Therefore, future research should focus more on a user-centred approach, which investigated in the users' needs, expectations, and prior knowledge. This could be in comprehension to a more intuitive and insightful design in XAI interfaces. Applying XAI techniques in more diverse real-work tasks, especially those with ethical or societal implications are crucial for providing insight in the strengths and limitations of existing methods. Even though some methods work well in easy made-up tasks does not necessarily mean that they serve the same benefits in a more complex task. Therefore, future work should also focus in comparing XAI methods based on more diverse tasks to get a better understanding in where the limitations of the XAI method are and to determine if the task itself poses challenges.

11. References

[1] J. Garland, K. Ghazi-Zahedi, J.-G. Young, L. Hébert-Dufresne, and M. Galesic, 'Countering hate on social media: Large scale classification of hate and counter speech'. arXiv, Jun. 05, 2020. Accessed: Jan. 29, 2023. [Online]. Available: http://arxiv.org/abs/2006.01974

[2] G. I. Pérez-Landa, O. Loyola-González, and M. A. Medina-Pérez, 'An Explainable Artificial Intelligence Model for Detecting Xenophobic Tweets', *Appl. Sci.*, vol. 11, no. 22, p. 10801, Nov. 2021, doi: 10.3390/app112210801.

[3] J. Cobbe, 'Algorithmic Censorship by Social Platforms: Power and Resistance', *Philos. Technol.*, vol. 34, no. 4, pp. 739–766, Dec. 2021, doi: 10.1007/s13347-020-00429-0.

[4] X. Yu, E. Blanco, and L. Hong, 'Hate Speech and Counter Speech Detection: Conversational Context Does Matter', in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States: Association for Computational Linguistics, 2022, pp. 5918–5930. doi: 10.18653/v1/2022.naacl-main.433.

[5] P. Blackman Sphaier and A. Paes, 'User intent classification in noisy texts: an investigation on neural language models', *Neural Comput. Appl.*, vol. 34, no. 20, pp. 17381–17406, Oct. 2022, doi: 10.1007/s00521-022-07383-2.

[6] T. Ahmed, S. Ivan, M. Kabir, H. Mahmud, and K. Hasan, 'Performance analysis of transformer-based architectures and their ensembles to detect trait-based cyberbullying', *Soc. Netw. Anal. Min.*, vol. 12, no. 1, p. 99, Dec. 2022, doi: 10.1007/s13278-022-00934-4.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. arXiv, May 24, 2019. Accessed: Jan. 30, 2023. [Online]. Available: http://arxiv.org/abs/1810.04805

[8] Md. R. Karim *et al.*, 'DeepHateExplainer: Explainable Hate Speech Detection in Under-resourced Bengali Language', in *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, Porto, Portugal: IEEE, Oct. 2021, pp. 1–10. doi: 10.1109/DSAA53316.2021.9564230.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, 'Model-Agnostic Interpretability of Machine Learning', 2016, doi: 10.48550/ARXIV.1606.05386.

[10] S. Sikdar, P. Bhattacharya, and K. Heese, 'Integrated Directional Gradients: Feature Interaction Attribution for Neural NLP Models', in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online: Association for Computational Linguistics, Aug. 2021, pp. 865–878. doi: 10.18653/v1/2021.acl-long.71.

[11] A. Modarressi, M. Fayyaz, Y. Yaghoobzadeh, and M. T. Pilehvar, 'GlobEnc: Quantifying Global Token Attribution by Incorporating the Whole Encoder Layer in Transformers', in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 258–271. doi: 10.18653/v1/2022.naacl-main.19.

[12] A. Das, C. Gupta, V. Kovatchev, M. Lease, and J. J. Li, 'ProtoTEx: Explaining Model Decisions with Prototype Tensors'. arXiv, May 22, 2022. doi: 10.48550/arXiv.2204.05426.

[13] G. Attanasio, D. Nozza, E. Pastor, and D. Hovy, 'Benchmarking Post-Hoc Interpretability Approaches for Transformer-based Misogyny Detection', in *Proceedings of NLP Power! The First Workshop on Efficient Benchmarking in NLP*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 100–112. doi: 10.18653/v1/2022.nlppower-1.11.

[14] Z. Wu and D. C. Ong, ‘On Explaining Your Explanations of BERT: An Empirical Study with Sequence Classification’. arXiv, Jan. 01, 2021. doi: 10.48550/arXiv.2101.00196.
[15] F. Bodria, A. Panisson, A. Perotti, and S. Piaggesi, ‘Explainability Methods for Natural Language Processing: Applications to Sentiment Analysis (Discussion Paper)’.
[16] S. Krishna *et al.*, ‘The Disagreement Problem in Explainable Machine Learning: A Practitioner’s Perspective’. arXiv, Feb. 08, 2022. Accessed: Mar. 25, 2023. [Online]. Available: http://arxiv.org/abs/2202.01602
[17] O. Lampridis, L. State, R. Guidotti, and S. Ruggieri, ‘Explaining short text classification with diverse synthetic exemplars and counter-exemplars’, *Mach. Learn.*, May 2022, doi: 10.1007/s10994-022-06150-7.
[18] Z. Sourati *et al.*, ‘Robust and Explainable Identification of Logical Fallacies in Natural Language Arguments’. arXiv, Jan. 27, 2023. Accessed: Jul. 22, 2023. [Online]. Available: http://arxiv.org/abs/2212.07425
[19] M. Sundararajan, A. Taly, and Q. Yan, ‘Axiomatic Attribution for Deep Networks’. arXiv, Jun. 12, 2017. Accessed: Mar. 23, 2023. [Online]. Available: http://arxiv.org/abs/1703.01365
[20] A. Jacovi and Y. Goldberg, ‘Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness?’, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, 2020, pp. 4198–4205. doi: 10.18653/v1/2020.acl-main.386.
[21] P. Lopes, E. Silva, C. Braga, T. Oliveira, and L. Rosado, ‘XAI Systems Evaluation: A Review of Human and Computer-Centred Methods’, *Appl. Sci.*, vol. 12, no. 19, p. 9423, Sep. 2022, doi: 10.3390/app12199423.
[22] S. Mohseni, N. Zarei, and E. D. Ragan, ‘A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems’, *ACM Trans. Interact. Intell. Syst.*, vol. 11, no. 3–4, pp. 1–45, Dec. 2021, doi: 10.1145/3387166.
[23] E. Cambria, L. Malandri, F. Mercorio, M. Mezzanzanica, and N. Nobani, ‘A survey on XAI and natural language explanations’, *Inf. Process. Manag.*, vol. 60, no. 1, p. 103111, Jan. 2023, doi: 10.1016/j.ipm.2022.103111.
[24] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, ‘Metrics for Explainable AI: Challenges and Prospects’. arXiv, Feb. 01, 2019. Accessed: Jul. 16, 2023. [Online]. Available: http://arxiv.org/abs/1812.04608
[25] B. Mathew *et al.*, ‘Thou shalt not hate: Countering Online Hate Speech’. arXiv, Apr. 04, 2019. Accessed: Jan. 29, 2023. [Online]. Available: http://arxiv.org/abs/1808.04409
[26] B. Mathew, N. Kumar, Ravina, P. Goyal, and A. Mukherjee, ‘Analyzing the hate and counter speech accounts on Twitter’, 2018, doi: 10.48550/ARXIV.1812.02712.
[27] Q. Li *et al.*, ‘A Survey on Text Classification: From Traditional to Deep Learning’, *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 2, pp. 1–41, Apr. 2022, doi: 10.1145/3495162.
[28] Y. Zhang, R. Jin, and Z.-H. Zhou, ‘Understanding bag-of-words model: a statistical framework’, *Int. J. Mach. Learn. Cybern.*, vol. 1, no. 1–4, pp. 43–52, Dec. 2010, doi: 10.1007/s13042-010-0001-0.
[29] W. Cavnar and J. Trenkle, ‘N-Gram-Based Text Categorization’, *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, May 2001.
[30] 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: Association for Computational Linguistics, 2014, pp. 1532–1543. doi: 10.3115/v1/D14-1162.

[31] ‘Rule-Based Classifier - Machine Learning’, *GeeksforGeeks*, May 06, 2020. https://www.geeksforgeeks.org/rule-based-classifier-machine-learning/ (accessed Aug. 31, 2023).
[32] M. E. Maron, ‘Automatic Indexing: An Experimental Inquiry’, *J. ACM*, vol. 8, no. 3, pp. 404–417, Jul. 1961, doi: 10.1145/321075.321084.
[33] T. Joachims, ‘Text categorization with support vector machines’, Universität Dortmund, Oct. 1999. doi: 10.17877/DE290R-5097.
[34] T. Cover and P. Hart, ‘Nearest neighbor pattern classification’, *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967, doi: 10.1109/TIT.1967.1053964.
[35] A. Gasparetto, A. Zangari, M. Marcuzzo, and A. Albarelli, ‘A survey on text classification: Practical perspectives on the Italian language’, *PLOS ONE*, vol. 17, no. 7, p. e0270904, Jul. 2022, doi: 10.1371/journal.pone.0270904.
[36] S. Albawi, T. A. Mohammed, and S. Al-Zawi, ‘Understanding of a convolutional neural network’, in *2017 International Conference on Engineering and Technology (ICET)*, Antalya: IEEE, Aug. 2017, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.
[37] S. Pouyanfar *et al.*, ‘A Survey on Deep Learning: Algorithms, Techniques, and Applications’, *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–36, Sep. 2019, doi: 10.1145/3234150.
[38] S. Lyu and J. Liu, ‘Combine Convolution with Recurrent Networks for Text Classification’. arXiv, Jun. 28, 2020. Accessed: Jul. 27, 2023. [Online]. Available: http://arxiv.org/abs/2006.15795
[39] W. K. Sari, D. P. Rini, and R. F. Malik, ‘Text Classification Using Long Short-Term Memory With GloVe Features’, *J. Ilm. Tek. Elektro Komput. Dan Inform.*, vol. 5, no. 2, p. 85, Feb. 2020, doi: 10.26555/jiteki.v5i2.15021.
[40] A. Madsen, ‘Visualizing memorization in RNNs’, *Distill*, vol. 4, no. 3, p. 10.23915/distill.00016, Mar. 2019, doi: 10.23915/distill.00016.
[41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, ‘Learning representations by back-propagating errors’, *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.
[42] ‘Backpropagation | Brilliant Math & Science Wiki’. https://brilliant.org/wiki/backpropagation/ (accessed Sep. 03, 2023).
[43] Mazur, ‘A Step by Step Backpropagation Example’, *Matt Mazur*, Mar. 17, 2015. https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/ (accessed Sep. 03, 2023).
[44] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
[45] M. Schuster and K. K. Paliwal, ‘Bidirectional recurrent neural networks’, *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997, doi: 10.1109/78.650093.
[46] R. Pascanu, T. Mikolov, and Y. Bengio, ‘On the difficulty of training Recurrent Neural Networks’. arXiv, Feb. 15, 2013. Accessed: Jul. 27, 2023. [Online]. Available: http://arxiv.org/abs/1211.5063
[47] K. Cho *et al.*, ‘Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation’. arXiv, Sep. 02, 2014. Accessed: Aug. 31, 2023. [Online]. Available: http://arxiv.org/abs/1406.1078
[48] D. Bahdanau, K. Cho, and Y. Bengio, ‘Neural Machine Translation by Jointly Learning to Align and Translate’. arXiv, May 19, 2016. Accessed: Jul. 27, 2023. [Online]. Available: http://arxiv.org/abs/1409.0473
[49] ‘Attention in Neural Networks - 8. Alignment Models (1) · Buomsoo Kim’. https://buomsoo-kim.github.io/attention/2020/03/05/Attention-mechanism-8.md/ (accessed Aug. 01, 2023).
[50] K. Xu *et al.*, ‘Show, Attend and Tell: Neural Image Caption Generation with Visual Attention’. arXiv, Apr. 19, 2016. Accessed: Jul. 27, 2023. [Online]. Available: http://arxiv.org/abs/1502.03044
[51] T. Luong, H. Pham, and C. D. Manning, ‘Effective Approaches to Attention-based Neural Machine Translation’, in *Proceedings of the 2015 Conference on Empirical*

Methods in Natural Language Processing, Lisbon, Portugal: Association for Computational Linguistics, 2015, pp. 1412–1421. doi: 10.18653/v1/D15-1166.
[52] J. Cheng, L. Dong, and M. Lapata, 'Long Short-Term Memory-Networks for Machine Reading', in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, 2016, pp. 551–561. doi: 10.18653/v1/D16-1053.
[53] 'What is the intuition behind the dot product attention?', *Educative: Interactive Courses for Software Developers*. https://www.educative.io/answers/what-is-the-intuition-behind-the-dot-product-attention (accessed Sep. 01, 2023).
[54] A. Vaswani *et al.*, 'Attention Is All You Need', *ArXiv170603762 Cs*, Dec. 2017, Accessed: Nov. 27, 2020. [Online]. Available: http://arxiv.org/abs/1706.03762
[55] S. Kamat, 'What Is Tokenization For NLP?', *Data, AI & Product Engineering | Algoscale*, Jun. 08, 2022. https://algoscale.com/blog/what-is-tokenization-for-nlp/ (accessed Sep. 01, 2023).
[56] 'Scaled Dot-Product Attention', *OpenGenus IQ: Computing Expertise & Legacy*, May 01, 2023. https://iq.opengenus.org/scaled-dot-product-attention/ (accessed Sep. 02, 2023).
[57] '11.1. Queries, Keys, and Values — Dive into Deep Learning 1.0.3 documentation'. https://d2l.ai/chapter_attention-mechanisms-and-transformers/queries-keys-values.html (accessed Sep. 02, 2023).
[58] J. Alammar, 'The Illustrated Transformer'. http://jalammar.github.io/illustrated-transformer/ (accessed Sep. 02, 2023).
[59] S. Cristina, 'The Transformer Attention Mechanism', *MachineLearningMastery.com*, Sep. 14, 2022. https://machinelearningmastery.com/the-transformer-attention-mechanism/ (accessed Aug. 05, 2023).
[60] R. Kulshrestha, 'Transformers', *Medium*, Nov. 22, 2020. https://towardsdatascience.com/transformers-89034557de14 (accessed Sep. 04, 2023).
[61] *Transformer Neural Networks - EXPLAINED! (Attention is all you need)*, (Jan. 13, 2020). Accessed: Sep. 04, 2023. [Online Video]. Available: https://www.youtube.com/watch?v=TQQlZhbC5ps
[62] S. Chandak, 'BERT for Multi-class text classification', *Medium*, Jul. 20, 2019. https://medium.com/@chandaksumit29_15695/bert-for-multi-class-text-classification-12b66a1fc01c (accessed Sep. 04, 2023).
[63] 'BERT — transformers 2.11.0 documentation'. https://huggingface.co/transformers/v2.11.0/model_doc/bert.html (accessed Aug. 12, 2023).
[64] 'BERT Tokenization', Mar. 14, 2023. https://tinkerd.net/blog/machine-learning/bert-tokenization/ (accessed Aug. 12, 2023).
[65] 'MLM — Sentence-Transformers documentation'. https://www.sbert.net/examples/unsupervised_learning/MLM/README.html (accessed Jul. 29, 2023).
[66] Y. Zhu *et al.*, 'Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books'. arXiv, Jun. 22, 2015. Accessed: Jun. 05, 2023. [Online]. Available: http://arxiv.org/abs/1506.06724
[67] S. Sharma, S. Sharma, and A. Athaiya, 'ACTIVATION FUNCTIONS IN NEURAL NETWORKS', *Int. J. Eng. Appl. Sci. Technol.*, vol. 04, no. 12, pp. 310–316, May 2020, doi: 10.33564/IJEAST.2020.v04i12.054.
[68] H. Chefer, S. Gur, and L. Wolf, 'Transformer Interpretability Beyond Attention Visualization', presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 782–791. Accessed: Jan. 17, 2023. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Chefer_Transformer_Interpretability_Beyond_Attention_Visualization_CVPR_2021_paper.html

[69] S. Chakraborty *et al.*, ‘Interpretability of deep learning models: A survey of results’, in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, San Francisco, CA: IEEE, Aug. 2017, pp. 1–6. doi: 10.1109/UIC-ATC.2017.8397411.
[70] F. Doshi-Velez and B. Kim, ‘Towards A Rigorous Science of Interpretable Machine Learning’. arXiv, Mar. 02, 2017. Accessed: Jan. 29, 2023. [Online]. Available: http://arxiv.org/abs/1702.08608
[71] A. B. Arrieta *et al.*, ‘Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI’. arXiv, Dec. 26, 2019. Accessed: Jan. 29, 2023. [Online]. Available: http://arxiv.org/abs/1910.10045
[72] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, ‘Explainable AI: A Review of Machine Learning Interpretability Methods’, *Entropy*, vol. 23, no. 1, p. 18, Dec. 2020, doi: 10.3390/e23010018.
[73] G. Vilone and L. Longo, ‘Notions of explainability and evaluation approaches for explainable artificial intelligence’, *Inf. Fusion*, vol. 76, pp. 89–106, Dec. 2021, doi: 10.1016/j.inffus.2021.05.009.
[74] European Commission. Joint Research Centre., *Robustness and explainability of Artificial Intelligence: from technical to policy solutions.* LU: Publications Office, 2020. Accessed: Mar. 25, 2023. [Online]. Available: https://data.europa.eu/doi/10.2760/57493
[75] E. Galinkin, ‘Robustness and Usefulness in AI Explanation Methods’. arXiv, Mar. 07, 2022. Accessed: Mar. 25, 2023. [Online]. Available: http://arxiv.org/abs/2203.03729
[76] T. Speith, ‘A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods’, in *2022 ACM Conference on Fairness, Accountability, and Transparency*, Seoul Republic of Korea: ACM, Jun. 2022, pp. 2239–2250. doi: 10.1145/3531146.3534639.
[77] G. Schwalbe and B. Finzel, ‘A Comprehensive Taxonomy for Explainable Artificial Intelligence: A Systematic Survey of Surveys on Methods and Concepts’, *Data Min. Knowl. Discov.*, Jan. 2023, doi: 10.1007/s10618-022-00867-8.
[78] A. Saranya and R. Subhashini R, ‘A systematic review of Explainable Artificial Intelligence models and applications: Recent developments and future trends’, *Decis. Anal. J.*, vol. 7, p. 100230, Jun. 2023, doi: 10.1016/j.dajour.2023.100230.
[79] W. Saeed and C. Omlin, ‘Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities’, *Knowl.-Based Syst.*, vol. 263, p. 110273, Mar. 2023, doi: 10.1016/j.knosys.2023.110273.
[80] A. Adadi and M. Berrada, ‘Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)’, *IEEE Access*, vol. 6, pp. 52138–52160, 2018, doi: 10.1109/ACCESS.2018.2870052.
[81] C. Molnar, *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*, 2nd ed. 2023. Accessed: Aug. 06, 2023. [Online]. Available: https://christophm.github.io/interpretable-ml-book/
[82] C. Munoz, K. da Costa, B. Modenesi, and A. Koshiyama, ‘Local and Global Explainability Metrics for Machine Learning Predictions’. arXiv, Feb. 23, 2023. Accessed: Aug. 13, 2023. [Online]. Available: http://arxiv.org/abs/2302.12094
[83] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell, ‘What do we need to build explainable AI systems for the medical domain?’ arXiv, Dec. 28, 2017. Accessed: Aug. 13, 2023. [Online]. Available: http://arxiv.org/abs/1712.09923
[84] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen, ‘A Survey of the State of Explainable AI for Natural Language Processing’. arXiv, Oct. 01, 2020. doi: 10.48550/arXiv.2010.00711.
[85] A. M. P. Brasoveanu and R. Andonie, ‘Visualizing Transformers for NLP: A Brief Survey’, in *2020 24th International Conference Information Visualisation (IV)*,

Melbourne, Australia: IEEE, Sep. 2020, pp. 270–279. doi: 10.1109/IV51561.2020.00051.
[86] K. Simonyan, A. Vedaldi, and A. Zisserman, 'Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps'. arXiv, Apr. 19, 2014. Accessed: Mar. 23, 2023. [Online]. Available: http://arxiv.org/abs/1312.6034
[87] S. Lundberg and S.-I. Lee, 'A Unified Approach to Interpreting Model Predictions'. arXiv, Nov. 24, 2017. Accessed: Jan. 29, 2023. [Online]. Available: http://arxiv.org/abs/1705.07874
[88] X. Jin, Z. Wei, J. Du, X. Xue, and X. Ren, 'Towards Hierarchical Importance Attribution: Explaining Compositional Semantics for Neural Sequence Models'. arXiv, Jun. 15, 2020. doi: 10.48550/arXiv.1911.06194.
[89] J. Li, X. Chen, E. Hovy, and D. Jurafsky, 'Visualizing and Understanding Neural Models in NLP'. arXiv, Jan. 08, 2016. Accessed: Aug. 13, 2023. [Online]. Available: http://arxiv.org/abs/1506.01066
[90] P.-J. Kindermans *et al.*, 'The (Un)reliability of saliency methods'. arXiv, Nov. 02, 2017. doi: 10.48550/arXiv.1711.00867.
[91] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, 'On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation', *PLOS ONE*, vol. 10, no. 7, p. e0130140, Jul. 2015, doi: 10.1371/journal.pone.0130140.
[92] S. Abnar and W. Zuidema, 'Quantifying Attention Flow in Transformers', in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, 2020, pp. 4190–4197. doi: 10.18653/v1/2020.acl-main.385.
[93] Z. Wu and D. C. Ong, 'On Explaining Your Explanations of BERT: An Empirical Study with Sequence Classification'. arXiv, Jan. 01, 2021. Accessed: Jan. 29, 2023. [Online]. Available: http://arxiv.org/abs/2101.00196
[94] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, 'SmoothGrad: removing noise by adding noise'. arXiv, Jun. 12, 2017. Accessed: Jul. 09, 2023. [Online]. Available: http://arxiv.org/abs/1706.03825
[95] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, 'Towards better understanding of gradient-based attribution methods for Deep Neural Networks'. arXiv, Mar. 07, 2018. Accessed: Jul. 09, 2023. [Online]. Available: http://arxiv.org/abs/1711.06104
[96] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, 'Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization', *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, Feb. 2020, doi: 10.1007/s11263-019-01228-7.
[97] S. Jain and B. C. Wallace, 'Attention is not explanation', in *Proceedings of the 2019 Conference of the North*, Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 3543–3556. doi: 10.18653/v1/N19-1357.
[98] S. Wiegreffe and Y. Pinter, 'Attention is not not Explanation', 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: Association for Computational Linguistics, 2019, pp. 11–20. doi: 10.18653/v1/D19-1002.
[99] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, 'Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned', in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, 2019, pp. 5797–5808. doi: 10.18653/v1/P19-1580.
[100] I. Tenney, D. Das, and E. Pavlick, 'BERT Rediscovers the Classical NLP Pipeline', in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, 2019, pp. 4593–4601. doi: 10.18653/v1/P19-1452.

[101] Y. Hao, L. Dong, F. Wei, and K. Xu, ‘Self-Attention Attribution: Interpreting Information Interactions Inside Transformer’. arXiv, Feb. 25, 2021. doi: 10.48550/arXiv.2004.11207.
[102] E. Voita, R. Sennrich, and I. Titov, ‘The Bottom-up Evolution of Representations in the Transformer: A Study with Machine Translation and Language Modeling Objectives’, 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: Association for Computational Linguistics, 2019, pp. 4395–4405. doi: 10.18653/v1/D19-1448.
[103] S. Eger, J. Daxenberger, and I. Gurevych, ‘How to Probe Sentence Embeddings in Low-Resource Languages: On Structural Design Choices for Probing Task Evaluation’. arXiv, Oct. 28, 2020. Accessed: Sep. 06, 2023. [Online]. Available: http://arxiv.org/abs/2006.09109
[104] J. Vig, ‘Visualizing Attention in Transformer-Based Language Representation Models’. arXiv, Apr. 11, 2019. Accessed: Jan. 29, 2023. [Online]. Available: http://arxiv.org/abs/1904.02679
[105] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, ‘What Does BERT Look at? An Analysis of BERT’s Attention’, in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 276–286. doi: 10.18653/v1/W19-4828.
[106] B. van Aken, B. Winter, A. Löser, and F. A. Gers, ‘VisBERT: Hidden-State Visualizations for Transformers’. arXiv, Nov. 09, 2020. doi: 10.48550/arXiv.2011.04507.
[107] B. Hoover, H. Strobelt, and S. Gehrmann, ‘exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformers Models’. arXiv, Oct. 11, 2019. doi: 10.48550/arXiv.1910.05276.
[108] B. Škrlj, N. Eržen, S. Sheehan, S. Luz, M. Robnik-Šikonja, and S. Pollak, ‘AttViz: Online exploration of self-attention for transparent neural language modeling’. arXiv, May 12, 2020. Accessed: Mar. 23, 2023. [Online]. Available: http://arxiv.org/abs/2005.05716
[109] G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui, ‘Attention is Not Only a Weight: Analyzing Transformers with Vector Norms’. arXiv, Oct. 06, 2020. Accessed: Jan. 31, 2023. [Online]. Available: http://arxiv.org/abs/2004.10102
[110] B. van Aken, B. Winter, A. Löser, and F. A. Gers, ‘VisBERT: Hidden-State Visualizations for Transformers’, in *Companion Proceedings of the Web Conference 2020*, Taipei Taiwan: ACM, Apr. 2020, pp. 207–211. doi: 10.1145/3366424.3383542.
[111] D. Nozza, F. Bianchi, and D. Hovy, ‘What the [MASK]? Making Sense of Language-Specific BERT Models’. arXiv, Mar. 05, 2020. Accessed: Sep. 06, 2023. [Online]. Available: http://arxiv.org/abs/2003.02912
[112] S. Velampalli, C. Muniyappa, and A. Saxena, ‘Performance Evaluation of Sentiment Analysis on Text and Emoji Data Using End-to-End, Transfer Learning, Distributed and Explainable AI Models’, *J. Adv. Inf. Technol.*, vol. 13, no. 2, 2022, doi: 10.12720/jait.13.2.167-172.
[113] G. Ansari, P. Kaur, and C. Saxena, ‘Data Augmentation for Improving Explainability of Hate Speech Detection’, *Arab. J. Sci. Eng.*, Jul. 2023, doi: 10.1007/s13369-023-08100-4.
[114] H. Sebbaq and N. El Faddouli, ‘MTBERT-Attention: An Explainable BERT Model based on Multi-Task Learning for Cognitive Text Classification’, *Sci. Afr.*, vol. 21, p. e01799, Sep. 2023, doi: 10.1016/j.sciaf.2023.e01799.
[115] H. Mehta and K. Passi, ‘Social Media Hate Speech Detection Using Explainable Artificial Intelligence (XAI)’, *Algorithms*, vol. 15, no. 8, p. 291, Aug. 2022, doi: 10.3390/a15080291.
[116] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, ‘Not Just a Black Box: Learning Important Features Through Propagating Activation Differences’. arXiv,

Apr. 11, 2017. Accessed: Sep. 06, 2023. [Online]. Available: http://arxiv.org/abs/1605.01713
[117] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, 'A Survey of Methods for Explaining Black Box Models', *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, Sep. 2019, doi: 10.1145/3236009.
[118] A. Das, 'ProtoTEx: Explaining Model Decisions with Prototype Tensors'. Dec. 18, 2022. Accessed: Aug. 05, 2023. [Online]. Available: https://github.com/anubrata/ProtoTEx
[119] O. Li, H. Liu, C. Chen, and C. Rudin, 'Deep Learning for Case-Based Reasoning Through Prototypes: A Neural Network That Explains Its Predictions', *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2018, doi: 10.1609/aaai.v32i1.11771.
[120] M. Lewis *et al.*, '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: Association for Computational Linguistics, 2020, pp. 7871–7880. doi: 10.18653/v1/2020.acl-main.703.
[121] 'cross-encoder/nli-deberta-base · Hugging Face', Jan. 25, 2023. https://huggingface.co/cross-encoder/nli-deberta-base (accessed Sep. 08, 2023).
[122] 'typeform/distilbert-base-uncased-mnli · Hugging Face', Apr. 05, 2023. https://huggingface.co/typeform/distilbert-base-uncased-mnli (accessed Sep. 08, 2023).
[123] 'howey/electra-base-mnli · Hugging Face'. https://huggingface.co/howey/electra-base-mnli (accessed Sep. 08, 2023).
[124] 'cross-encoder/nli-roberta-base · Hugging Face', Jan. 25, 2023. https://huggingface.co/cross-encoder/nli-roberta-base (accessed Sep. 08, 2023).
[125] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, 'ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators'. arXiv, Mar. 23, 2020. Accessed: Sep. 08, 2023. [Online]. Available: http://arxiv.org/abs/2003.10555
[126] J. Bastings and K. Filippova, 'The elephant in the interpretability room: Why use attention as explanation when we have saliency methods?', in *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, Online: Association for Computational Linguistics, Nov. 2020, pp. 149–155. doi: 10.18653/v1/2020.blackboxnlp-1.14.
[127] S. Liu, F. Le, S. Chakraborty, and T. Abdelzaher, 'On Exploring Attention-based Explanation for Transformer Models in Text Classification', in *2021 IEEE International Conference on Big Data (Big Data)*, Dec. 2021, pp. 1193–1203. doi: 10.1109/BigData52589.2021.9671639.
[128] 'Captum · Model Interpretability for PyTorch'. https://captum.ai/ (accessed Aug. 02, 2023).
[129] 'Alibi Explain — Alibi 0.9.5dev documentation'. https://docs.seldon.io/projects/alibi/en/latest/index.html (accessed Aug. 02, 2023).
[130] F. Friedrich, P. Schramowski, C. Tauchmann, and K. Kersting, 'Interactively Providing Explanations for Transformer Language Models'. arXiv, Mar. 11, 2022. doi: 10.48550/arXiv.2110.02058.
[131] 'Counterfactual explanation for text classification — OmniXAI documentation'. https://opensource.salesforce.com/OmniXAI/latest/tutorials/nlp/ce_classification.html (accessed Sep. 09, 2023).
[132] S. Feng, E. Wallace, A. Grissom Ii, M. Iyyer, P. Rodriguez, and J. Boyd-Graber, 'Pathologies of Neural Models Make Interpretations Difficult', in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 3719–3728. doi: 10.18653/v1/D18-1407.
[133] 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: Association for Computational Linguistics, Nov. 2020, pp. 3256–3274. doi: 10.18653/v1/2020.emnlp-main.263.
[134] M. Ivanovs, R. Kadikis, and K. Ozols, 'Perturbation-based methods for explaining deep neural networks: A survey', *Pattern Recognit. Lett.*, vol. 150, pp. 228–234, Oct. 2021, doi: 10.1016/j.patrec.2021.06.030.
[135] M. Nagahisarchoghaei *et al.*, 'An Empirical Survey on Explainable AI Technologies: Recent Trends, Use-Cases, and Categories from Technical and Application Perspectives', *Electronics*, vol. 12, no. 5, p. 1092, Feb. 2023, doi: 10.3390/electronics12051092.
[136] S. Liu, Z. Li, T. Li, V. Srikumar, V. Pascucci, and P.-T. Bremer, 'NLIZE: A Perturbation-Driven Visual Interrogation Tool for Analyzing and Interpreting Natural Language Inference Models', *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 651–660, 2019, doi: 10.1109/TVCG.2018.2865230.
[137] E. Wallace, J. Tuyls, J. Wang, S. Subramanian, M. Gardner, and S. Singh, 'AllenNLP Interpret: A Framework for Explaining Predictions of NLP Models'. arXiv, Sep. 19, 2019. doi: 10.48550/arXiv.1909.09251.
[138] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, 'Gradient-Based Attribution Methods', in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, Eds., in Lecture Notes in Computer Science, vol. 11700. Cham: Springer International Publishing, 2019, pp. 169–191. doi: 10.1007/978-3-030-28954-6_9.
[139] J. L. Kolodner, 'An introduction to case-based reasoning', *Artif. Intell. Rev.*, vol. 6, no. 1, pp. 3–34, 1992, doi: 10.1007/BF00155578.
[140] 'Hugging Face – The AI community building the future.', Aug. 04, 2023. https://huggingface.co/ (accessed Aug. 05, 2023).
[141] K. Safjan, 'LIME - Understanding How This Method for Explainable AI Works', *Krystians Safjan Blog*, 2023.
[142] M. Neely, S. F. Schouten, M. J. R. Bleeker, and A. Lucic, 'Order in the Court: Explainable AI Methods Prone to Disagreement'. arXiv, Jul. 06, 2021. Accessed: Jul. 09, 2023. [Online]. Available: http://arxiv.org/abs/2105.03287
[143] S. Benesch, 'Countering Dangerous Speech: New Ideas for Genocide Prevention', *SSRN Electron. J.*, 2014, doi: 10.2139/ssrn.3686876.
[144] C. Buerger, 'Counterspeech: A Literature Review', *SSRN Electron. J.*, 2021, doi: 10.2139/ssrn.4066882.
[145] C. Schieb and M. Preuss, 'Governing hate speech by means of counterspeech on Facebook', Jun. 2016.
[146] J. Barlett and A. Krasodomski-Jones, 'counter-speech examining content that challenges extremism online', Oct. 2015. https://demos.co.uk/wp-content/uploads/2015/10/Counter-speech.pdf (accessed Jul. 22, 2023).
[147] R. Frenett and M. Dow, 'One to One Online Interventions – A Pilot CVE Methodology', *ISD*. https://www.isdglobal.org/isd-publications/one-to-one-online-interventions-a-pilot-cve-methodology/ (accessed Jul. 22, 2023).
[148] J. Miškolci, L. Kováčová, and E. Rigová, 'Countering Hate Speech on Facebook: The Case of the Roma Minority in Slovakia', *Soc. Sci. Comput. Rev.*, vol. 38, no. 2, pp. 128–146, Apr. 2020, doi: 10.1177/0894439318791786.
[149] P. Koehn, 'Europarl: A Parallel Corpus for Statistical Machine Translation', in *Proceedings of Machine Translation Summit X: Papers*, Phuket, Thailand, Sep. 2005, pp. 79–86. [Online]. Available: https://aclanthology.org/2005.mtsummit-papers.11
[150] P. Koehn, 'Europarl (European Parliament Proceedings Parallel Corpus)', 2005. https://www.statmt.org/europarl/index.html (accessed Jul. 22, 2023).
[151] 'AdamW — PyTorch 2.0 documentation'. https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html (accessed Aug. 05, 2023).

[152]‘Trainer’. https://huggingface.co/docs/transformers/main_classes/trainer (accessed Aug. 05, 2023).
[153]‘PyTorch’. https://www.pytorch.org (accessed Aug. 06, 2023).
[154]‘Google Colaboratory’. https://colab.research.google.com/ (accessed Aug. 06, 2023).
[155]M. T. C. Ribeiro, ‘lime’. Aug. 03, 2023. Accessed: Aug. 05, 2023. [Online]. Available: https://github.com/marcotcr/lime
[156]M. Ribeiro, S. Singh, and C. Guestrin, “‘Why Should I Trust You?”: Explaining the Predictions of Any Classifier’, in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, San Diego, California: Association for Computational Linguistics, 2016, pp. 97–101. doi: 10.18653/v1/N16-3020.
[157]‘zhpinkman/Prototype-Learning at 55301a9e3eded177f7190b04c39f705f6dbfeac0’, *GitHub*. https://github.com/zhpinkman/Prototype-Learning (accessed Aug. 05, 2023).
[158]M. Fayyaz, ‘GlobEnc’. May 03, 2023. Accessed: Aug. 06, 2023. [Online]. Available: https://github.com/mohsenfayyaz/GlobEnc
[159]M. Sokolova and G. Lapalme, ‘A systematic analysis of performance measures for classification tasks’, *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, Jul. 2009, doi: 10.1016/j.ipm.2009.03.002.
[160]‘sklearn.dummy.DummyClassifier’, *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.dummy.DummyClassifier.html (accessed Sep. 08, 2023).
[161]S. Turney, ‘Pearson Correlation Coefficient (r) | Guide & Examples’, *Scribbr*, May 13, 2022. https://www.scribbr.com/statistics/pearson-correlation-coefficient/ (accessed Aug. 10, 2023).
[162]G. R. Loftus and M. E. J. Masson, ‘Using confidence intervals in within-subject designs’, *Psychon. Bull. Rev.*, vol. 1, no. 4, pp. 476–490, Dec. 1994, doi: 10.3758/BF03210951.
[163]G. Cumming, *Understanding The New Statistics: Effect Sizes, Confidence Intervals, and Meta-Analysis*, 1st ed. Routledge, 2013. doi: 10.4324/9780203807002.
[164]J. Perktold *et al.*, ‘statsmodels/statsmodels: Release 0.14.0’. Zenodo, May 05, 2023. doi: 10.5281/ZENODO.593847.
[165]P. Hamm, M. Klesel, P. Coberger, and H. F. Wittmann, ‘Explanation matters: An experimental study on explainable AI’, *Electron. Mark.*, vol. 33, no. 1, p. 17, Dec. 2023, doi: 10.1007/s12525-023-00640-9.
[166]T. Tullis and B. Albert, *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. in The Morgan Kaufmann interactive technologies series. Amsterdam ; Boston: Elsevier/Morgan Kaufmann, 2008.
[167]G. Singh, ‘ANOVA: Complete guide to Statistical Analysis & Applications (Updated 2023)’, *Analytics Vidhya*, Jan. 15, 2018. https://www.analyticsvidhya.com/blog/2018/01/anova-analysis-of-variance/ (accessed Aug. 04, 2023).

12. List of Tables

13. List of Figures