Master Thesis in Institute of Computer Science at Freie Universität Berlin

Data Science

Evaluating The Explanation of Black Box Decision for Text Classification

Esra GÜCÜKBEL

Supervisor: MSc. Manuel Heurich

First Reviewer: Prof. Dr. Eirini Ntoutsi

Second Reviewer: Prof. Dr. Tim Landgraf

Berlin, March 28, 2023

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material that has been quoted either literally or by content from the used sources. The thesis was not examined before, nor has it been published. The submitted electronic version of the thesis matches the printed version.

Berlin, March 28, 2023

Esra GÜCÜKBEL

Abstract

Through progressively evolved technology, applications of machine learning and deep learning methods become prevalent with the increased size of the collected data and the data processing capacity. Among these methods, deep neural networks achieve high accuracy results in various classification tasks; nonetheless, they have the characteristic of opaqueness that causes called them black box models. As a trade-off, black box models fall short in terms of interpretability by humans. Without a supportive explanation of why the model reaches a particular conclusion, the output causes an intrusive situation for decision-makers who will take action with the outcome of predictions. In this context, various explanation methods have been developed to enhance the interpretability of black box models. LIME, SHAP, and Integrated Gradients techniques are examples of more adaptive approaches due to their well-developed and easy-to-use libraries. While LIME and SHAP are post-hoc analysis tools, Integrated Gradients provide model-specific outcomes using the model's inner workings. In this thesis, four widely used explanation methods are quantitatively evaluated for text classification tasks using the Bidirectional LSTM model and DistillBERT model on four benchmark data sets, such as SMS Spam, IMDB Reviews, Yelp Polarity, and Fake News data sets. The results of the experiments reveal that analysis methods and evaluation metrics provide an auspicious foundation for assessing the strengths and weaknesses of explanation methods.

Zusammenfassung

Durch die fortschreitende technologische Entwicklung werden Anwendungen des maschinellen Lernens und Deep-Learning-Methoden mit der zunehmenden Größe der gesammelten Daten und der Datenverarbeitungskapazität immer häufiger eingesetzt. Unter diesen Methoden erzielen tiefe neuronale Netze bei verschiedenen Klassifizierungsaufgaben eine hohe Genauigkeit; dennoch haben sie die Eigenschaft der Undurchsichtigkeit, die dazu führt, dass sie als Black-Box-Modelle bezeichnet werden. Im Gegenzug sind Blackbox-Modelle für den Menschen nur schwer interpretierbar. Ohne eine unterstützende Erklärung, warum das Modell zu einer bestimmten Schlussfolgerung gelangt, führt die Ausgabe zu einer unangenehmen Situation für Entscheidungsträger, die aufgrund der Vorhersagen Maßnahmen ergreifen werden. In diesem Zusammenhang sind verschiedene Erklärungsmethoden entwickelt worden, um die Interpretierbarkeit von Black-Box-Modellen zu verbessern. Die Techniken LIME, SHAP und Integrated Gradients sind Beispiele für adaptive Ansätze, da sie über gut entwickelte und einfach zu verwendende Bibliotheken verfügen. Als es sich bei LIME und SHAP um Post-hoc-Analysewerkzeuge handelt, liefern Integrated Gradients modellspezifische Ergebnisse unter Verwendung der inneren Funktionsweise des Modells. In dieser Arbeit werden vier weit verbreitete Erklärungsmethoden für Textklassifizierungsaufgaben unter Verwendung des bidirektionalen LSTM-Modells und des DistillBERT-Modells auf vier Benchmark-Datensätzen quantitativ evaluiert, wie z. B. SMS-Spam, IMDB-Rezensionen, Yelp-Polarität und Fake-News-Datensätze. Die Ergebnisse der Experimente zeigen, dass Analysemethoden und Bewertungsmetriken eine vielversprechende Grundlage für die Bewertung der Stärken und Schwächen von Erklärungsmethoden bieten.

Acknowledgments

I am always amazed by the data and the meaning we can create out of it. I would like to thank all my professors who taught me how to conduct research and apply meaning to the data. I would like to thank my supervisors, Prof. Dr. Eirini Ntoutsi, MSc. Manuel Heurich and MSc. Yi Cai, who gave me the opportunity to work in the field I wanted to conduct research and supported me through the whole thesis process.

Special thanks to my family and friends, who were always on my side during the tough times.

Contents

List of Figures

List of Tables

List of Algorithms

1. Introduction

1.1. Motivation

Deep learning models became widespread in recent years by quickly accessing high-performed hardware, transferable pre-trained models on Artificial Intelligence(AI) platforms, and the ease of integrating large models by commonly used frameworks. Object recognition, medical image segmentation to identify patterns in anatomical images, autonomous driving, and forecasting analysis for stock market data may be given to exemplify the fields of utilization. In parallel, the Natural Language Processing(NLP) domain has leaped forward by enormous text data on social platforms, forums, websites, and wikis. Hate Speech detection, speech recognition, sentiment analysis, topic categorization, named entity recognition, and text generation tasks compelled the attention of researchers in academia and professionals from the industry.

Although AI applications are evolving rapidly, providing highly accurate predictions and possessing extensive use cases, their implementation in the real world is spreading at a slower pace. The main reason is that the functioning of deep learning models is indeterminate and not fully accountable for the decisions. The fact that the factors on which the system's conclusion depends, its weaknesses, and possible biased results are implicit to direct interpretation by the model leads to these models being evaluated as *black-box* models and not being adopted in applications with critical consequences. With the intent of developing lawful, ethical and robust implementations, European Commission formalized "Ethics Guidelines for Trustworthy AI" [1]. Transparency is one of the core principles which requires explainability, traceability and auditability. While traceability refers to uncovering the inner functioning and decisions made by the system, explainability focus on technical processes and related human decisions. Auditability entails two of them since the system and results require openness all the time. The logging mechanism and evaluation reports facilitate transparency. Furthermore, the European Commission has guaranteed the rights of data subjects to obtain transparency and information on the logic of automated processing through the General Data Protection Regulation[2].

Interpretability comes into play to disperse the incompleteness and confirm the desiderata of robustness, privacy, fairness, causality and reliability of deep learning models [3]. Explanation methods are the proxy for the reasoning of the decisions by extracting rules, evaluating the attribute importances by perturbation, occlusion or influence, and intrinsically representing the model's

inner-working[4]. Even though explanation methods shed light on the model, they still require evaluation by the human examiner or functionally in many aspects, predominantly fidelity and correctness terms. Back propagated attribution methods [5] shows the limitations of attribution methods due to infidelity of feature importances, class insensitivity and difficulty of evaluation. Furthermore adversarial trained models [6] are able to deceive post-hoc explanation methods.

1.2. Contribution

Through the thesis, we strive to handle evaluating the explanation methods on different data sets by using an identical Deep Learning model. The aspects of the evaluation procedure and the contributions are listed below.

- **Measuring the ability of detecting important features**: We represent the outcomes of four explanation methods such as LIME[7], SHAP [8], Integrated Gradients[9] and Word Omission[10] techniques. The aim is to compare the ability to detect important features of widely used explanation methods using Area Over Perturbation Curve(AOPC), Log-Odds, novel Relative AOPC metrics on IMDB 50K, Yelp(sampled), Fake News and Spam Collection data sets. Furthermore, the setup allows us to benchmark[11] the outcomes within the IMDB 50K data set in terms of AOPC and Log-Odds metrics.
- **Identity property evaluation**: Identity property requires generating the same feature importances for the same instances when the explanation method is operated several times. It is introduced in [12] for tabular data sets and we provide the thruputs for text data sets.
- **Switching point metric**: Similarly to [13] by masking words in sequences until the predicted class is changed, the number of removed tokens is counted on a percentage basis. Despite the fact that the data sets are different, the technique enables us to compare the numbers with the result of the study.
- **Processing Time**: Even though processing time might be counted in Qualitative Evaluation, it is a notable factor whilst deciding upon the methods in real-world applications for decision makers, we measured the processing times of explanation methods for each individual instance of batches in all data sets.

1.3. Thesis Outline

The thesis mainly contains four sections. *Section 2: Related Work* involves a literature review on deep learning methods which take place in the scope of

Black Box models and Explanation Methods are explained to provide background before the implementation. Afterward, *Section 3: Implementation* provides general information such as the sizes, collection methodology, and annotations about the data sets which are given as input to the models and evaluation processes. The preferred methods for preprocessing are clarified. In the second part of this section, studies on LSTM and BERT model design, hyper-parameter optimization, and model performance are presented. *Section 4: Evaluation* is the backbone of the thesis, where the explanation methods described in *Section 2* are used to evaluate the interpretability of the black box model, both for simple texts and using automatic evaluation. Finally, *Section 5: Conclusion* summarizes the work and discusses further improvement points.

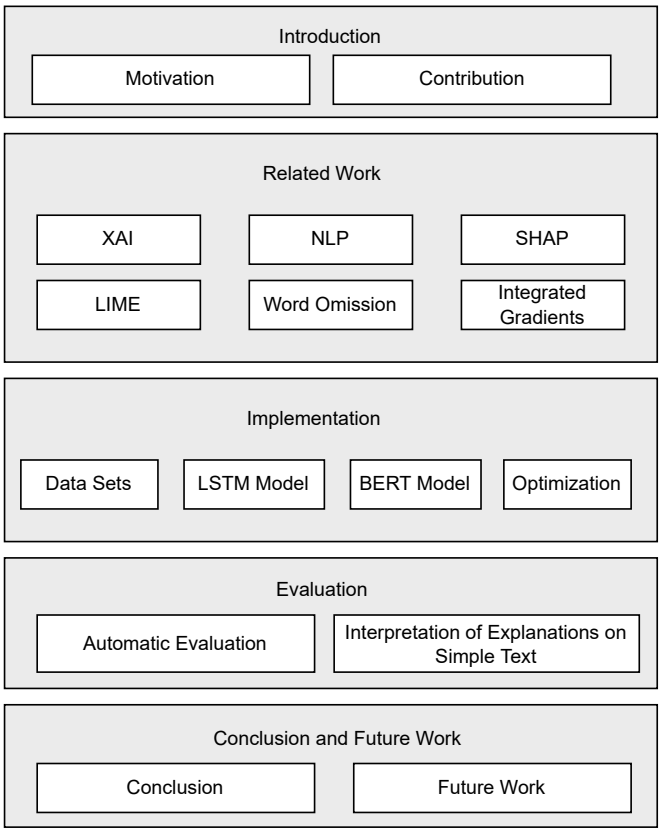

Figure 1.1.: Thesis Outline and Structure

2. Related Work

2.1. Deep Learning Models for Text Classification Tasks

In this section, we outline the deep learning models in two sections: Neural Networks which comprise Artificial, Convolutional and Recurrent Neural Networks and state-of-the-art Deep Language Models that consider the input words and phrases in their context. Ahead of mentioning the models, it is noteworthy to emphasize the nature of text data and transformation techniques.

Text data is inherently processed differently from images and numeric tabular data, which implicitly affects the architecture of the network. Therefore several transformation methods were developed to process text data in advance of the model-building phase. Bag of Words, Term Frequency-Inverse Document frequency(TF-IDF) as an extension of Bag Of Words, Count Vectorizer and Tokenization are the transformation approaches to transform the feature space(sentence, word or character form) into latent space(numeric vector form), which is taken as an input in order to be learned by the model or alternatively pre-trained word embeddings such as Glove[14], Word2Vec[15], FastText[16] might be adopted as embedding unit. The embedding layer maps the array of word indices to a dimensional vector. The input dimension is determined by vocabulary size, the output dimension corresponds to the embedding dimension and the input length is equal to the maximum length of the sentence that is specified through text-to-sequence transformation. The limitations of these methods which are vocabulary size, the maximum length of sentence, and preserving syntactic and semantic structure should be considered while selecting a transformation technique.

In the following part, the methods used for text classification will be presented in two sections, Neural Networks and Deep Language Model as similarly structured in [17].

2.1.1. Neural Networks

Artificial Neural Networks(ANN) are purely neuron-based models, which learn the weights of each feature in the input layer and output a score for each class. Regarding to the number of hidden layer, the architecture is qualified as a shallow or deep(multi-layer perceptron) neural network. Commonly the input layer consists of a bag of word vector representation, [18] showcase sentiment analysis using vector average of the unordered bag of words embedding, although it performs better in terms of time complexity, it suffers from

double negation due to lack of syntactic context. [19] labels the topics by using top 100 relevant candidates from doc2vec and word2vec neural embedding vectors.

Convolutional Neural Networks(CNN) are known as state-of-art architectures for the tasks of medical image segmentation[20], visual object recognition[21] in computer vision field due to their feature extraction capability by using convolution filters, in point of fact that they reveal remarkable performance in text categorization[22], sentiment analysis[23] tasks in Natural Language Processing(NLP) domain. Moreover, CNN models enable users to obtain faster and just as accurate outcomes without the necessity of going into more depth. [24] expanded upon a comparison of different architectures exploiting in NLP field, especially sentiment classification.[25] compared the accuracy of a simple multi-layer perceptron model using a bag of words embedding with state-of-art graph-based text classification methods on single and multi-class text classification tasks.

Recurrent Neural Networks(RNN) are predominantly used models for language translation[26], text classification[27], generating image captions[28], forecasting based on time-series data[29]. Its nature is well suited for sequential data which has characteristics of temporality. RNN units take prior inputs into consideration in order to produce results for current input, that called as 'memory' property of RNNs. There are different types RNNs for various purposes, such as many(input) to one(output) for text classification, many to many for language translation, etc. On the other hand, RNNs suffer from exploding and vanishing gradients since the weights might easily get close to zero or getting extremely large through the backpropagation process when updating weights. To overcome the vanishing and exploding gradients issue, the various architectures are derivated from RNNs, such as Long-Term Short Term Memory(LSTM), Gated Recurrent Units(GRU), Bi-directional RNN(Bi-RNN). [30] composes strengths of CNNs and LSTMs, where CNNs have the ability to extract features, and semantic structure is preserved on account of the LSTM part. [27] stacks multiple recurrent units to perform fine-grained sentiment analysis using binary parse trees, the results show that deep RNN obtains a different aspect of compositionality in each layer.

2.1.2. Deep Language Models

Encoder-Decoder Architecture using RNNs addresses the problem of fixed-length vector input and outputs, particularly an impediment for language translation [31] or sequence to sequence prediction [26]. While traditional neural networks require fixed-length input vectors and output vectors, Encode-Decoder architecture allows the generation of variable lengths due to discrete encoder and decoder models. The encoder layer maps the input sequence to a fixed-length vector and creates context information by extracting dependen-

cies(conditional probabilities) between words, decoder layer maps the encoded vector to target sequences [26].

Attention Mechanism has emerged as an enhancement to the drawback that context vectors, created by encoding the whole sentence into a fixed-length vector, transmit sparse information and do not give attention to critical words or pixels that should be concentrated on. The context vectors of target indexes rely on mapped annotations by the encoder layer and carry the information about the whole input. The vital part is estimating weights for input indexes since the context words correspond to the weighted sum of annotations. This model is called an "alignment model" by [31]. The alignment score represents straightforwardly the coherence of indexes around the input position in the encoding layer for the corresponding output position in the decoding layer. Different methods exist to utilize attention scores in deep neural networks, such as scaled-dot-product attention. Vaswani et al.[32] introduced the attention mechanism for language translation using scaled dot product attention scores due to providing faster and space-efficient computation. The attention mechanism has a wide range of usage in NLP tasks to improve the model accuracy and provide self-interpretability by emphasizing the specific parts that affected the outcome of the model's decision.

Transformer Architecture takes into account the information from different representation sub-spaces at different positions, a method called multi-head attention, instead of a single attention head depicted in Attention Mechanism. The main difference with other architectures, that the authors[33] described, is a self-attention layer, which applies positional encoding to sequences to find their absolute or relative positions right before the encoder and decoder layers. Bidirectional Encoder Representations(BERT)[34] from Transformers is pre-trained on two tasks, through the first task is called Masked Language Model, it masks a certain percentage of word tokens and predicts these tokens instead of constructing the sentence from scratch, for the second task called as Next Sentence Prediction tries to predict if next sentence B is actual next sentence of original sentence A. Pre-trained BERT model provides transfer learning by transmitting the model parameters to downstream tasks and uplifts the outcomes of many tasks in text domain. Analogously Generative Pre-Trained Transformer(GPT)[35] is an auto-regressive model that is pre-trained on next-word prediction using only decoders in the stacked architecture and enables the transfer learning, nonetheless distinctly from BERT, it learns from only one directional sequence, originally left to right.

2.2. Explainability and Interpretability

Black box models inherently have a mechanism where we know the output but do not know how and why it reaches the conclusion. Regression models or De-

cision Tree models are called algorithmically transparent models due to their ability to reproduce the output with the model's coefficients. On the contrary, as argued by Lipton et al.[36], the interpretability of algorithmically transparent models should not be accepted blindly, the associations between features could be deceptive depending on feature engineering and preprocessing. Furthermore, [3] supports this argument with the statement "Interpretability is used to confirm other important desiderata of ML systems". In the light of this information, interpretability is defined as the ability to explain or to present in understandable terms to a human[3].

Even though black box models e.g. random forest, ensemble models, neural networks especially deep learning algorithms, outperforms linear models in the context of numerous fields, scientists and decision-makers hesitate to deploy these models to real-world applications, especially in scenarios where the consequences could be severe disastrous[37]. Since blindly trust even 100% accurate model without knowing whether the model will perform correctly for unknown situations in the future. The models lacking in interpretability require a well-designed user-oriented proxy to make them perceivable by the human brain. Nevertheless recently self-interpretable models[38, 39] have been studied and explanation methods[7, 8, 9, 13, 40] to understand model's behavior globally and locally, by perturbing inputs or approximating them with surrogate models[41] have been introduced.

2.2.1. Taxonomy of Interpretability and Properties of Explanation Methods

The interpretability methods can be classified regarding three criteria, phase of the process, structural source and scope. The phase of the process includes three stages of model building, Pre-Model, In-Model and Post-Model[41]. Exploratory analysis and data visualization are given as examples of pre-model methods, in-model interpretability takes the inner workings of the model into consideration and post-model methods enable users to interpret the model after the training phase. Another criterion is the structural source of the method, intrinsic and post-hoc[42]. Intrinsic interpretability corresponds to the structure of the model as regards their complexities, and post-hoc interpretability is derived from the results of explanation methods after model training. Finally, there are two scopes of explanation, such as model-specific and model-agnostic. Model-specific interpretation is based on the model's internals, such as weights of the linear model, and decision points of the decision tree model, on the other hand, model-agnostic interpretability is a global approach for local instances, these techniques are advantageous for all ML models since they are applicable after model training and used for explaining instances.

From a broader perspective, the explanation may have characteristics of four

properties that are proposed by [43] to evaluate the explanation methods. In the first place, the representation style of explanation denotes expressive power. The following two properties are opposed to each other. Translucency requires model-specific explanation, so that focuses on the model's internal mechanism, oppositely portability property searches applications that are compatible with any possible model. Lastly, algorithmic complexity corresponds to computational time, the better explanation should be feasible even in deep ML models.

2.2.2. Properties of Instance Based Explanations

In this thesis, we will focus on instance-based explanations. In an attempt to construct a framework for whether an explanation reflects the model's behavior on seen and unseen data, its weak and strong points, [43] set forward some properties, yet as stated in [41] the utilization and benefits of these properties on specific use-cases are not clear.

- Accuracy: Explanation's competence of predicting accurately unseen data points. For example, reflecting inaccurate results of ML model thoroughly.

- Fidelity: The ability to explain the prediction of the ML model. High fidelity is desired, because if an explanation can not explain the prediction, then it doesn't provide correct information about the model.

- Consistency: Two different models are trained on the same tasks and they produce similar outputs. In that case, the feature importances of instances should be very similar. Nonetheless, if the models use different features during training but still give similar feature importances, that situation is undesirable. It relies on similar feature importances for similar features on different models for the same data set.

- Stability(Robustness): Whilst two instances are similar, in other words slightly different, the explanation method should give similar results. However in the text domain, replacing one word with another is sufficient to interchange the sentiment, therefore this axiom needs to be examined in detail. Furthermore, the sampling phase of some methods is another factor to might prevents ensuring this property.

- Certainty: This property is related to ML model, providing the confidence score about the prediction.

- Comprehensibility: The understandability and interpretability of explanations by humans. It requires a human examiner in an experiment context, therefore is hard to evaluate this property.

- Importance: Representation style of importance in feature set. The importance of features should be significant and emphasized.

- Novelty: It diagnoses the strength of the method in case of encountering unseen data point which is distant from training data. If the model predicts this instance wrongfully, the most likely explanation method will present the wrong explanation.

- Representativeness(Coverage): The number of instances that are covered by explanation. It reflects the scope of the method. For example, model-agnostic methods mostly cover one instance, whilst model-specific methods cover every instance in the data set.

2.3. Explanation Methods in Text Classification

Explanation methods are the facilitator for providing interpretability to the model. [4] gathers the explanation methods under the following taxonomy: rule extraction, attribution, and intrinsic methods. Rule extraction methods deduct the rules from the decision-making process, attributions methods are based on perturbing the inputs and measuring the variance between pre and post-modification, and intrinsic methods ensure interpretive capabilities through the process inherently, thus that might be informative visuals or loss function.

The following sections describe the explanation methods that are employed in the context of this thesis. We chose four different techniques from attribution methods to compare and contrast them on four data sets. LIME and SHAP as model-agnostic tools are widely used in applications, when LIME explains instances by perturbing the input, SHAP takes the marginal contribution of features into consideration. On the other hand, Integrated Gradients are an interpretability technique from model-specific methods, which rely on the gradients of the model. Furthermore, the word omission technique similar to occlusion in the image field obtains feature importance by removing words and calculating variance for individual features. The easiness of configuration and having different inner workings of these explanation methods are the factors that influence the selection.

2.3.1. Local Interpretable Model-Agnostic Explanations(LIME)

Local Interpretable Model-Agnostic Explanations are designed for explaining feature importance by using perturbations of instances. Local fidelity is taken as the key metric for determining the success of the explanation, the expectation is to create meaningful explanations for instances given to the model in order to understand the model behavior. It was introduced for classification models in [7] and we will explain the algorithm according to the reference. We use original LIME implementation[44] from its authors.

The main idea of LIME is perturbing the instance x and creating an interpretable model $g \in G$ that is the class of interpretable models to explain the features which is called the Sparse Linear Model. Let f represent the black box model and $f(x)$ may be probabilities with respect to the classes or binary-valued output vector(if $y_i \in$ y is 1, then the decision belongs to the respective class). $\Omega(g)$ represents the complexity of the explanation model so that it can be chosen as the number of non-zero weights for linear models. $\pi_x(z)$ corresponds to the weights of model g, since it measures the closeness of vectors x and z that is perturbed version of it. Last the term $\mathcal{L}(f, g, \pi_x)$ that is called "locality-aware loss" in the Equation 2.1 indicates the distance between the outputs of g and f. Through the optimization process, total loss ($\mathcal{L}$ and π_x) will be minimized.

$$\xi(x) := \underset{g \in G}{\operatorname{argmin}} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2.1}$$

$$\mathcal{L}(f, g, \pi_x) := \sum_{z, z' \in \mathcal{Z}} \pi_x(z)(f(z) - g(z'))^2 \tag{2.2}$$

In advance of sampling, the features of x are transformed into binary vector space that is called an interpretable vector representation. For example, x has n features and in the original version, in the original instance, we see all partitions of the vector is 1 because all of them were included, however, if a partition is changed by perturbation, its value turned to 0.

The essential step of the explanation process is forming up the samples. The samples are drawn uniformly at random around x' by taking the distance(closeness) into consideration, if an instance is close to x', it will have higher weight(π_x), otherwise lower weight. The data set $\mathcal{Z}$ of g is built with these samples. The goal is to find best-fitted weights w_g which mimic the behavior of base model f and minimize the loss between outputs of g and f. It is worth noting that if the base model is highly non-linear in predictions, then that may lead to the explanation model never converging to the base model.

The authors of [7] describe the desired characteristics for explainers, these are ensuring interpretable explanations, providing local fidelity, being used as a model-agnostic tool and global perspective. The first characteristic takes the limitations of the audience into consideration such as invested time, the number of features willingly checked, etc. The second characteristic is being meaningful and faithful at least on the local instance level, it corresponds to accuracy and fidelity properties in instance-based explanations. Methods may be a model agnostic tool if it is "plug and explain" to any model. Finally, the last characteristic is similar to the second one in terms of considering the

limitations of the audience, but distinctly, an explainer should also provide a global perspective by using expressive features to give an understanding to the user.

2.3.2. Shapley Additive Explanations(SHAP)

Shapley Additive Explanations was first introduced by L. S. Shapley in 1953 [45]. The idea behind it is the contribution of each player to the game and their rewards regarding to their performances. Hence, this logic is implemented for explaining the black box models to extract the feature importances.

The importance values are calculated in that way, the contribution values for the subset of features, in respect to the order of the features in the sentence, in which feature i exists and not exists are calculated, then the weighted marginal contribution for the feature is estimated with using Equation 2.3. S represents the subsets excluded i and F represents all features. The feature importances sum up to the prediction value.

$$\phi_i(v, F) := \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{F!} [v(S \cup \{i\}) - v(S)] \tag{2.3}$$

Nonetheless, the computation time is very long due to calculating all possible combinations, for example for the number of F features, a 2^F calculation should be made. [42]. As a consequence of the fact that approximate methods based on Shapley values,e.g Owen values for text domain, needed to be preferred to reduce computation time [46]. The SHAP library[47] enables the users' different versions of SHAP, such as Kernel SHAP, Tree SHAP, Deep SHAP and *Partitition Explainer* for NLP tasks that what we will focus on.

Partition Explainer takes Owen values as basis on[46], the difference from Shapley values is partitioning the sequences and creating coalitions(token groups) which may be correlated, instead of all combinations of the individual tokens in the sequence. In Equation 2.4[48][49], B represents all partitions for N features, k is the indexes of unions where i exists and M=1,2,3,..,m is quotient set of each B so that m denotes the sub-union of B. Q corresponds to the union of all unions where i not exists in these unions and T corresponds to a sub-union(coalition). As a result of the partition mechanism, the computation costs reduce up to 2^{b_k+m-1}[46].

$$\phi_i(v, B) := \sum_{R \subseteq M \setminus \{k\}} \sum_{T \subseteq B_k \setminus \{i\}} \frac{1}{m b_k} \frac{1}{\binom{m-1}{r}} \frac{1}{\binom{b_k - 1}{t}} [v(Q \cup T \cup \{i\}) - v(Q \cup T)] \tag{2.4}$$

where $Q := \cup_{r \in R} B_r$

Shapley values are known as satisfying the 4 properties of the attribution method, they are called *Efficiency, Symmetry, Dummy and Additivity*[48]. Owen values provide 3 of them but Symmetry, since if two tokens contribute equally to coalition then the expectation is their feature importances should be the same, however differently structured partitions may cause differences. In brief, dummy property requires a feature doesn't affect to model then its contribution should be zero, efficiency(local accuracy) gives the cumulative marginal contributions of features are equal to the result of prediction in terms of probability($\sum_i^N \phi_i$) and finally additivity ensures the sum of Shapley values even the final output comes from two different intermediary results.

2.3.3. Integrated Gradients

The Integrated Gradients technique aims to extract the feature importances regarding their gradients by considering the *Sensitivity*, *Completeness*, and *Implementation Invariance* properties. It was proposed in [9] and the implementation from Alibi Explain library[50] was used in this thesis.

According to [9], *sensitivity* requires the non-zero attribution value if the prediction probability changes when a feature differs in both baseline $x^{'}$ and input x, this feature is violated in Rectified Linear Unit(ReLu) networks($relu(x) = max0, x$), because even though the input changes, the output could remain same and that causes similar back-propagation values. Another condition is *implementation invariance* is satisfied when the outputs of two different models are the same for all inputs, the attribution values therefore the gradients should be the same as well, yet Layer-wise propagation and Deep LIFT substitute the formula of chain-rule $\frac{\partial f}{\partial g} = \frac{\partial f}{\partial h} \cdot \frac{\partial h}{\partial g}$ with discrete values $\frac{f(x_1)-f(x_0)}{h(x_1)-h(x_0)} \cdot \frac{h(x_1)-h(x_0)}{g(x_1)-g(x_0)}$ and these discrete values don't correspond to $\frac{f(x_1)-f(x_0)}{g(x_1)-g(x_0)}$. Lastly, *completeness* indicates the sum of attribution values should provide the equality of the difference between the output of baseline $x^{'}$ and input x.

Let's say F is a function that denotes the black box model's decision. $R^n \implies [0, 1]$ for each input x which comprises of $[x_1, x_2, x_3, ..., x_n] \in R^n$. x_i symbolizes attributes or in other words features of the input. The attribution values of x with respect to baseline input $x^{'}$ is a vector $A_F(x, x^{'}) = [x_1, x_2, x_3, ..., x_n]$. Baseline vectors are mostly chosen as zeros that correspond to the index value of padding (<PAD>) for text-related tasks and black pixels for image-related tasks.

The technique computes the gradients taking a straight line path from input x to baseline $x^{'}$ in consideration. In the text domain, we have 1 dimension but for multi-dimensional problem spaces, the gradients of $F(x)$ in the i^{th} dimension

are defined as $\frac{\partial F(x)}{\partial x_i}$. The Equation 2.5 shows the calculation of integrated gradients values in the i^{th} dimension which is proposed in [9].

$$IntegratedGrads_i(x) := (x_i - x_i^i) \times \int_{\alpha=0}^{1} \frac{\partial F(x^{'} + \alpha \times (x - x^{'}))}{\partial x_i} d\alpha \quad (2.5)$$

In order to approximate by summing up the integrated gradient values in m steps using the Riemann approximation method, Equation 2.6 is used.

$$IntegratedGrads_i^{approx}(x) := (x_i - x_i^i) \times \sum_{k=1}^{m} \frac{\partial F(x^{'} + \frac{k}{m} x(x - x^{'}))}{\partial x_i} \times \frac{1}{m} \quad (2.6)$$

2.3.4. Word Omission

Word omission is a very simplistic but effective approach to determining the word-level feature importances, similar to the occlusion method in image domain[51, 52]. It measures the change in the strength of probability for the predicted class by removing the word from the sentence, changing the word index to <UNK> or <PAD>, else changing the vector of the word with full zeros[10] in case of using TF-IDF or Bag-of-Word embeddings. This approach is proposed in [13] to compare the explanation methods which comprise our baseline, and it originates from [53] including log-odds based information difference in class.

In our approach, all sentences were transformed into sequences, each word index is set to 0(<pad>) respectively in sequence, then the probability for the altered sequence is calculated and the difference is taken as a feature importance value. The implementation of the word omission approach in our processes is given in Algorithm 2 in Appendix A.

2.4. Evaluation of Explanations

Thus far four explanation methods being used in text classification and the requested properties of instance-based explanations have been introduced. The explanation methods strive to decipher the model's decision-making. Therefore they may have different positive and negative aspects which need to be evaluated in depth. Even though there is no structured guideline or consensus upon evaluation, as proposed in [41], we can group them under *Qualitative* and *Quantitative* approaches. The metrics specified below correspond to the requested properties given in Section 2.2.2.

Qualitative approaches mostly match with human-centered applications. [3] lays out Application-Grounded and Human-Grounded evaluations in this group.

The application-Grounded evaluation aims to conduct experiments on real tasks with real users who will benefit from the outputs of the model in real life, while Human grounded evaluation focuses on simplified tasks. The main differentiation is that the user will actually operate the application and has therefore an expert in that field. The indicators(metrics)[3, 54, 55] related with qualitative approach are listed below:

- *Time*: Duration of the user is willing to analyze explanation results.
- *Number of units*: Depends on the time, and vice versa, how many units the user would like to observe in a certain time. The number of units, depending on the time that the user volunteer is taken into account in [7, 8] and interpreted as a parameter in the algorithms.
- *Form of units*: The form of units(atomic component) the user figures out, e.g word, raw data, image patch, etc.
- *Level of compositionality*: Are units grouped and formed up in compositions? SHAP' s cluster plots[56] included clustering of features can be given as an example.
- *Monotonicty and other interactions between units*: In which way are the units combined, linear or non-linear? Does the function matter to the users?
- *Uncertainty and stochasticity*: How does the user tackle uncertainty in explanation, is it obvious and understandable to by user?
- *Ease of use and configuration*: How easy is the path of setting(configuring) up and running the application of explanations?

Quantitative approaches corresponds to Functionally-Grounded applications of [3]. The human examiner doesn't involve evaluating the task, predefined functions or surrogate models are used as proxies to compare the performance. [12] proposed three axioms, *Identity, Separability, Stability* to evaluate the Consistency property using tabular data set and other research[57] aims to encapsulate evaluation in terms of three Cs, *Correctness, Completeness, Compactness*. They consist of many mutual points, particularly regarding Fidelity and Accuracy properties. Below the axioms from the two related research papers, are merged according to their similarity.

1. *Correctness*: The explanations should reflect the truth(accuracy) about the instance, in other words, they should be correct and ensure local fidelity. There are two axioms that are introduced by [12] within this framework.

a) *Identity*: If two instances are identically the same, then their explanations of them should be exactly the same. In case of not providing this property by method, implicitly accuracy property is not ensured.

b) *Separability*: Although two different instances have the same output in terms of probability or class-wise, their feature importances should not be the same.

2. *Stability*: The evaluation of Stability property in Section 2.2.

3. *Completeness*: The method should be able to generate the number of explanations as same as the number of instances.

4. *Compactness*: The methods should present concise results to the user, therefore it is a degree of complexity and directly affects the user's preferences while choosing the explanation method.

In addition to the indicators of the quantitative approach, automatic evaluation methods enable the users to evaluate the explanation method's competency in "fidelity and accuracy" properties by removing the important features. "Area Over Perturbation Curve(AOPC)" [40, 10], "Switching Points"[13], "Log-Odds"[11] is the automatic evaluation methods for evaluating the correctness of explanation.

2.4.1. Automatic Evaluation

The application of explanation techniques may vary according to ground truth or predicted class. For example, in the case where the model's output and the actual class are different, the explanation method produces results based on why the model is explained in this way. From this point of view, correctly and incorrectly classified examples should be analyzed. Similarly, as favored in our baseline work[13], separately examining the attributes that contribute positively and negatively to the outcome, depending on the model, gives more meaningful explanations for the outcome of the annotation.

Area Over Perturbation Curve(AOPC)

In order to measure local fidelity, the first N important features are removed from the text and change in probability with respect to the first predicted class is examined[40]. If the explanation method assigns large importance to the features, removing these features causes a drastic decrease in probability. As stated above, the type of evidence also plays an important role in the change of probability. If positive evidences are removed from the text, then obviously a decrease is expected, however, if we remove negative evidences

then the strength of the predicted class should increase.

The Equation 2.9 is applied to a set of samples, the value is matching up with the ability to fetch important features. For positive evidences, we expect high values, on the other hand for negative features smaller values are anticipated.

$$AOPC := \frac{1}{N}\sum_{i=1}^{N} p(\hat{y}|\hat{x}_i) - p(\hat{y}|\hat{x}_{i\setminus 1\ldots k}) \tag{2.7}$$

2.4.2. Relative AOPC

Relative AOPC is derived from a novel metric from wisely used metric AOPC, since whilst comparing the outcomes of explanation methods on one instance, if the methods find out the same features as important with different magnitudes, the AOPC function gives the same results due to masking the same features in sequence. By the means of Relative AOPC, we can observe the weighted difference between probabilities. N represents the number of instances, k is the number of selected features and m is the total number of features.

$$rAOPC := \sum_{i=1}^{N} \frac{|\sum_{i=1}^{k} f_{x_i}|}{|\sum_{i=1}^{m} fx_i|} * AOPC_i \tag{2.8}$$

2.4.3. Log-Odds

Similarly to the AOPC score, Log-Odds calculates the average natural logarithm of the probabilistic ratio before and after K selected features are masked on the text as described in [11].

$$LogOdds := \frac{1}{N}\sum_{i=1}^{N} log\frac{p(\hat{y}|\hat{x}_{i\setminus 1\ldots k})}{p(\hat{y}|\hat{x}_i)} \tag{2.9}$$

2.4.4. Switching Point

The switching point is another way to figure out an understanding explanation model's success. As its very name signifies, the logic behind is substantially deleting important features until the predicted class turns to another class.

Normalizing the output percentage over all the number of features in a sentence is taken, so that fewer words to turn to class demonstrate better explanation. The potential issue with this logic is that even though all words are removed from the text, the class of modified text may not be changed, particularly that can occur in short-length texts. The Algorithm A is provided in Appendix A.

2.4.5. Identity Score

As mentioned in the "Quantitative Approach" part, [12] proposed *Identity Score* for evaluating the consistency of the explanation methods. If there is a difference between two samples, intuitively the same sample, the explanations should be the same. Therefore we generate the feature importances from an explanation method twice for selected samples and compare the importance values of features whether they are the same and the percentage of features that do not have the same feature importances are estimated.

$$IdentityScore := \frac{\sum_{i}^{N} \frac{\sum_{j}^{m} (x_{aj} <> x_{bj})}{m}}{N} \tag{2.10}$$

3. Implementation

In this section, as a preliminary part of model design, the transformation of text to sequence stage is described, in the following the model design and optimization process are explained and the data sets are introduced.

3.1. Data Sets

In this study, one black box model is built and optimized for four different data sets. With this design, four highly polarized and binary classified data sets are used to evaluate the explanation methods. We consider selecting two large and two small data sets with different sentence lengths and part of speech distributions in data sets.

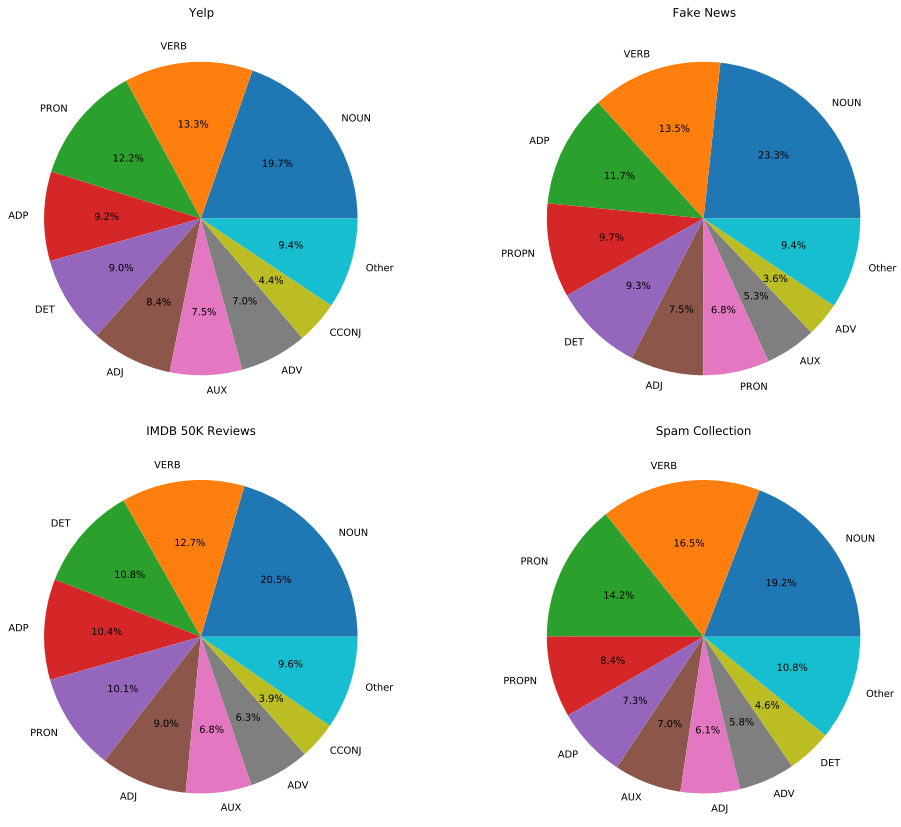

Figure 3.1.: The Distribution of Part of Speech Tags in Data Sets

The data sets have different characteristics and require to be explored to op-

timize the model. The sentence length has an impact on the model since the maximum length is a parameter for deciding upon the cut point of padding converted sequences. The details about the conversion process from feature space to latent space will be expanded upon in the section 3.2. The distribution of sentence lengths is presented in Figure 3.2.

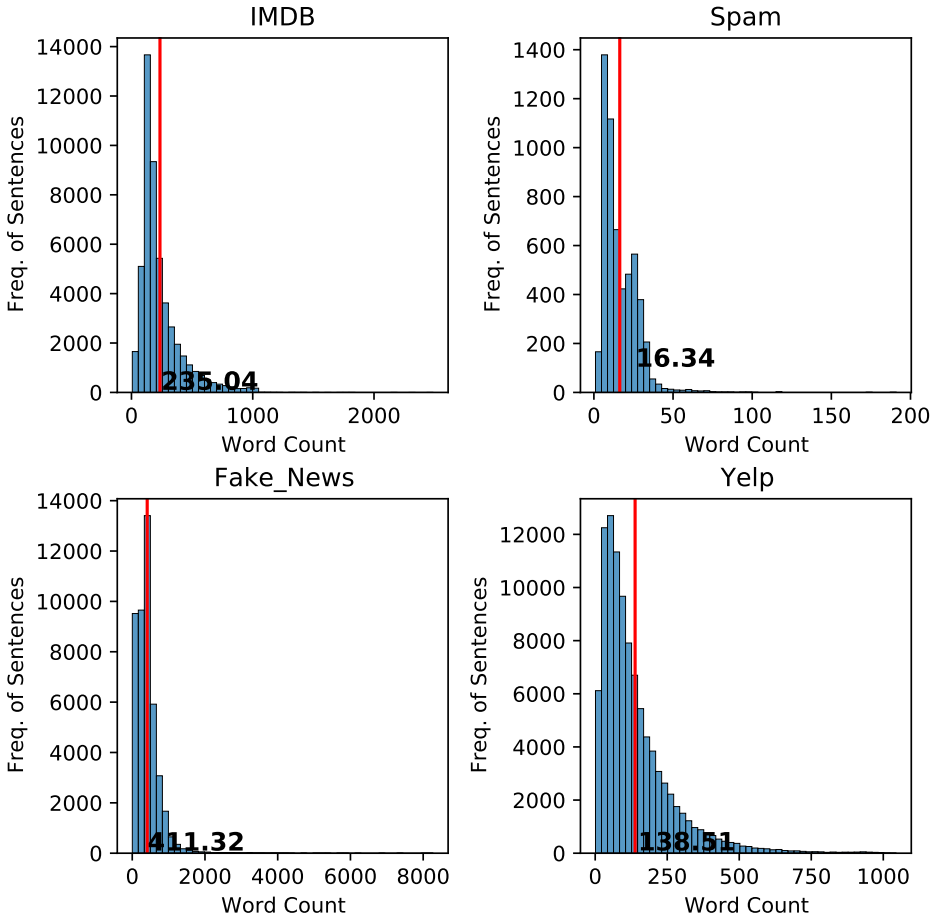

Figure 3.2.: The Distribution of Sentence Lengths in Data Sets

In an effort to standardize the input, data preprocessing techniques given below are implemented in four data sets. While reading data sets the column names are standardized as “text” and “label”, the letter cases are normalized, and special characters and HTTP tags are removed.

3.1.1. SMS Spam Data Set

SMS Spam Data Set is a collection of three sources. 425 Spam messages are collected from the Grumbletext Web site, 3725 ham messages are randomly chosen from the SMS Corpus of the Department of Computer Science at the National University of Singapore and 450 ham messages are picked up from Caroline Tagg's Ph.D. thesis. The data set and detailed information[58] are accessible through UCI archive [59].

Label	Name	Size
0	Ham	4825
1	Spam	747

Table 3.1.: SMS Spam Data Set

3.1.2. IMDB 50000 Data Set

The data set is constructed within the scope of capturing semantic similarities among words and determining the word level sentiments [60]. The data set consists of highly polarized reviews, the reviews which have above 4 out of 10 were labeled as negative and above 7 were labeled as positive. Originally data set was divided equally into training and test data sets, but we exploited the merged version that is accessible through the Kaggle[61]

Label	Name	Size
0	Negative	25000
1	Positive	25000

Table 3.2.: IMDB Data Set

3.1.3. Yelp Polarity Data Set

The data set is constructed for the "Advances in Neural Information Processing Systems 28" conference by [62]. It is a subset of the Yelp Dataset Challenge 2015 data set which has high polarity reviews. The data set consists of 560.000 training and 38.000 test samples. 1 and 2-star ratings are considered as the negative class, 3 and 4-star ratings as the positive class. We have sampled 100.000 instances from the train data set by setting the random state to 42 and the data source is accessible through Kaggle [63].

Label	Name	Size
0	Negative	42993
1	Positive	49926

Table 3.3.: Yelp Polarity Data Set

3.1.4. Fake News Data Set

The data set is collected for detecting online fake news using n-gram analysis[64]. It consists of true news from Reuters and truthful articles, fake news from Kaggle[65] and Politifact website.

Label	Name	Size
0	Negative	21417
1	Positive	23481

Table 3.4.: Fake News Data Set

3.2. Process Design

The thesis aims to evaluate the explanation methods in terms of competence in capturing important features of the decision. The process consists of mainly 4 parts, preprocessing of text data for being transformed into sequences, model building and predictions, generating feature importances from explanation methods, and evaluation of the results. The design of processes is depicted in Figure 3.3.

LSTM and BERT deep learning models are chosen in an effort to perform the text classification tasks. Model-specific preferences are clarified in related Sections 3.2.1,3.2.2 and the result is given in Section 3.2.3. LIME, SHAP, Integrated Gradients and Word Omission as explanation methods which are explained in Section 2.3 utilized to interpret the black box models and generate feature importances. The evaluation is examined thoroughly in Chapter 4.

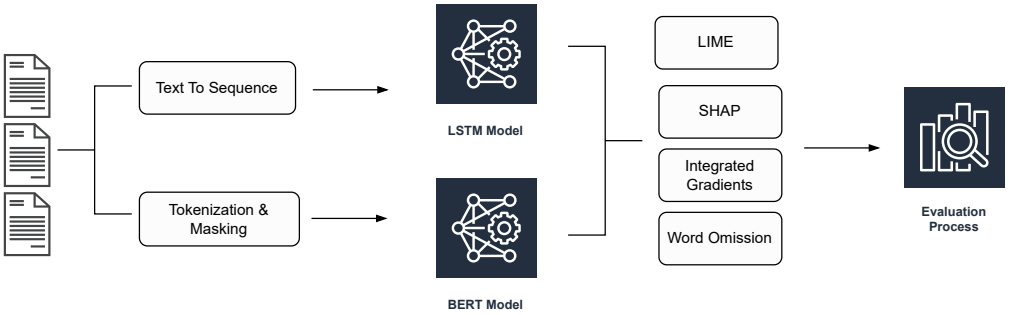

Figure 3.3.: The architecture of evaluation process

The maximum length varies in all data sets, hence it requires setting the inclusive and definitive maximum length of the data sets for input vectors given to the model. The table 3.5 represents the maximum number of tokens to pad the sequences. The maximum length hyper-parameter is kept fixed in both two models.

SMS Spam	**IMDB**	**Yelp Polarity**	**Fake News**
100	250	300	400

Table 3.5.: Maximum length hyper-parameter of the data sets

3.2.1. LSTM Model

As a core model, we employed the bidirectional LSTM model, since LSTM is capable of capturing long-term dependencies and the bidirectional version enhances the sequential learning process by using both directions of sentences. The *text_to_sequence* function included in the Tokenizer class of the Keras framework is employed to convert each token in a sentence into a sequence of integers. A dictionary is created to store these indexes and their word equivalents. The unknown words, out of vocabulary, are assigned to <unk> token. A noteworthy point about this process is that the dictionary is created for all unique tokens, not for the specified number of words given to Tokenizer for limiting the dictionary, but the encoding of new documents is applied for only the specified number of words, therefore the dictionary should be cut off the point of the number of words to not being in discordance with and the index 1 should be assigned <unk> token.

The model consists of respectively embedding layer, 64-unit bi-directional layer, 128-unit dense layer and lastly 1-unit dense layer. We train the model using the Adam optimizer with initial learning rate $l = 1e^{-3}$, epsilon $\epsilon = 1e^{-8}$, beta 1 β_1=0.9, beta 2 β_2=0.999, decay = 0 in 10 epochs. The batch size is set to 64 and the embedding dimension as the output of the embedding layer is

set to 100. In the following paragraphs, the layers used in this architecture are described thoroughly.

- **Embedding layer** is the first layer of the network which takes transformed vectors by tokenization and produces a dimension of ($vocabulary\_size$ x $maximum\_length$) vector for each output unit. Output units are basically dense units. In this thesis, we have preferred to use the data set's unique words as trainable vectors rather than using a pre-trained vector set.

- **Bi-directional LSTM layer** is the core part of the model. It consists of 64 units with all other settings in default values such as activation is hyperbolic tangent (tanh), no dropout for recurrent tensors, no regularizer and glorot uniform as kernel initializer.

- **Dense layer** is the $n-unit(neuron)$ layers associated with the activation function which bridges between feature map and output labels. The feature map passes through a fully connected (flattened) layer and the activation function classifies labels.

3.2.2. BERT Model

In the first place as a second model to compare model successes and explanation outputs, we have utilized an uncased BERT model with one hidden layer(512 units dense) and one output(1 unit dense) layer. It required learning 109M parameters and approximately takes 25 minutes for the training of the IMDB data set(40000 instances for training). Due to time and memory efficiency, distilled BERT uncased model is preferred. As stated in Transformers Documentation[66], it performs training with 40% fewer parameters than BERT uncased model and runs 60% faster while still retaining success over 95%. The model is straightforwardly instantiated for training without appending any specific layer, therefore loss function estimates the loss by using logits, nonetheless, the softmax function is appended to the trained model as the final layer to normalize output values. The training process yields results approximately in 15 minutes(for the IMDB dataset with 40000 instances) with very high accuracy values, the table 3.6 shows the training and test accuracy results.

Similarly to the LSTM model, the text should be transformed into tensors and further attention masks. Identical data pre-processing steps are implemented for cleaning text beforehand the transformation into tensors. Adam is chosen as the optimizer which set down the learning rate regarding the polynomial decay function and sparse categorical entropy for measuring the loss. Batch sizes are set to 16 due to memory limitations. The same maximum document length hyperparameters are used to clip or pad each sample.

3.2.3. Training and Test Results

Both two deep learning models are trained and operated on Google Colab virtual servers. 12GB NVIDIA Tesla K80 GPU and 13 GB RAM are utilized for high memory required operations, particularly whereas maximum sentence length is set to higher values. 20% of the data sets for the LSTM model and 25% of the data sets for the BERT model are split up as test sets. All data sets for the LSTM model are trained in 10 epochs, preliminary epoch number for the BERT model is set according to the total test size over batch size. With regard to train and test processes, the size of the train and test accuracies along with the size of data sets demonstrated in Table 3.6. DistillBERT model uplifts the accuracies by 5% for the IMDB data set and 3% for the Yelp data set.

Class	**LSTM**				**DistillBERT**			
	Spam	IMDB	Yelp	Fake News	Spam	IMDB	Yelp	Fake News
Train Set	4421	39970	40000	35394	4145	37500	37509	33182
Test Set	1106	9993	10000	8849	1382	12500	12491	11061
Train Acc.	100.0%	99.5%	99.3%	99.9%	97.2%	89.2%	93.1%	99.58%
Test Acc.	98.8%	86.4%	92.8 %	99.9%	98.9%	91.6%	95.5%	99.89%

Table 3.6.: The summary of model train and test processes

4. Evaluation of Explanation Methods

In this chapter, we present the interpretations of four explanation methods for one positive and one negative review from IMDB data sets in the first place, the related outputs for other data sets can be found in the Appendix 5.2 section. In the second section, the evaluations for four data sets are evaluated using automatic evaluation metrics. In sub-section 4.2.1, the explanations are demonstrated based on the outputs of two models and assessed using three evaluation metrics based on different significant attribution counts. Furthermore, "Identity Property", "Switching Points" and "Processing Times" are evaluated concerning explanation methods based on the outcomes of the LSTM model in the following sub-sections.

4.1. Interpretation of Explanation Outcomes Based On Coherent Samples

In this section, we aim to provide an empirical qualitative evaluation from the researcher's point of view and visualize the feature importances for one positive and one negative sample for the IMDB data set based on each explanation method's output in order to compare them side by side for LSTM and DistillBERT models.

Although we have qualitatively evaluated the methods of explanation here, it is worth emphasizing that this is only an interpretation from the researcher's perspective relying upon the experiments that are conducted through the thesis, and qualitative evaluations require a statistically significant sample of experts or general users depending on the content. The explanation methods, LIME [44], SHAP [47] and Integrated Gradients [50], are implemented by utilizing external libraries and the Word Omission and Random selection methods are developed by ourselves. The methods are qualitatively interpreted with reference to Section 2.4 below.

- Despite SHAP and LIME having visual representations of the data, Integrated Gradients and Word Omission require an intermediary step to visualize the results. The lack of simplicity in the setup process is a reference to the "Ease of use and configuration" property.

- LIME and SHAP enable the user to select the number of units to generate and visualize the attribute importances, other methods return the values for all units without any constraint.

- Explicitly the number of units affects the "Time" to spend willingly analyzing the data for the user(researcher or end-user), therefore SHAP and LIME allow the user to analyze the results effectively.

- The "Form of units" depends on the selected tokenization method, when LSTM is used as the model for explanation methods words are the smallest unit, however, DistillBERT tokenization splits words into segmented sub-words, hence it plays a significant role for all of the applied explanation methods.

- SHAP allows users to visualize the importance of feature clusters by considering the "Level of compositionality" for correlative features, nonetheless, other methods ensure the importance scores on a simple form of units.

- LIME and SHAP methods accommodate the randomness in the processes which may affect the reproducibility of the outcomes for samples, hence "stochasticity" property is evaluated thoroughly in Section 4.2. Nevertheless, the applied explanation methods do not provide a confidence score or range for explanations, therefore the evaluation of the "uncertainty" property is left out of scope.

We chose correctly classified one positive and one negative sample from the IMDB data set and perform the evaluation on LSTM and DistillBERT models to quantitatively compare the outputs of the explanation methods. The three most positively affected attributes are colored with orange hues, and the most negatively affected attributes with blue hues. Unlike other explanation methods, the Random selection method does not indicate the feature importance, since the method randomly selects words and their occurrences, therefore, it does not provide importance scores, the main purpose is to compare the selection outcomes of other methods with Random selection.

Table 4.1 presents the attribute importances for the sample which has negative sentiment. While using the BERT model, the positive contributing attributes *boring* are captured correctly by all methods in most of the cases, *slow* is detected by only the LIME method. Negative contributing attributes are incorrectly labeled, but only *awake* is labeled properly excluding Word Omission. On the contrary, while using the LSTM model, LIME could not detect *boring* and *slow* correctly, *awake* is not marked as a negative attribute in SHAP and Word Omission. The magnitude of attributes is higher for the DistillBERT model, in contrast to the positive sample.

Methods	LSTM	Distill BERT
Prediction Probabilities	[0.9999, 0.0001]	[0.947 0.052]
SHAP	*it* −0.0272 *is* −0.0459 *slow* 0.096 *and* −0.051 *boring* 0.259 *and* −0.057 *hard* 0.064 *for* −0.035 *me* −0.027 *to* 0.00964 *stay* 0.094 *awake* 0.065	*it* 0.044 *is* 0.040 *slow* 0.0121 *and* 0.018 *boring* 0.302 *and* −0.008 *hard* −0.004 *for* 0.036 *me* −0.041 *to* 0.032 *stay* 0.047 *awake* −0.010
LIME	*it* 0.0011 *is* 0.0042 *slow* −0.005 *and* 0.0036 *boring* −0.0163 *and* 0.0036 *hard* 0.0027 *for* 0.0038 *me* −0.00011 *to* 0.00071 *stay* −0.0071 *awake* −0.0039	*it* 0.035 *is* 0.061 *slow* 0.057 *and* −0.019 *boring* 0.448 *and* 0.001 *hard* 0.0032 *for* 0.041 *me* −0.0106 *to* 0.047 *stay* 0.009 *awake* −0.102
Integrated Gradients	*it* 0.004 *is* −0.069 *slow* 0.090 *and* −0.026 *boring* 0.126 *and* −0.013 *hard* 0.034 *for* −0.049 *me* 0.093 *to* 0.039 *stay* 0.123 *awake* −0.009	*it* −0.013 *is* −0.002 *slow* 0.023 *and* −0.028 *boring* 0.317 *and* −0.012 *hard* 0.008 *for* 0.086 *me* −0.015 *to* 0.063 *stay* 0.043 *awake* −0.029
Word Omission	*it* 0.00168 *is* 0.000329 *slow* 0.0025 *and* 0.00032 *boring* 0.030 *and* 0.00032 *hard* 0.00011 *for* 0.00004 *me* 0.00005 *to* 0.00010 *stay* 0.00036 *awake* 0.00014	*it* 0.022 *is* 0.055 *slow* −0.011 *and* 0.0109 *boring* 0.695 *and* −0.017 *hard* 0.080 *for* 0.083 *me* −0.032 *to* 0.209 *stay* 0.025 *awake* 0.0115
Random	*it* *is* *slow* *and* *boring* *and* *hard* *for* *me* *to* *stay* *awake*	*it* *is* *slow* *and* *boring* *and* *hard* *for* *me* *to* *stay* *awake*

Table 4.1.: Evaluation of explanations for simple text which is classified as negative review

Table 4.2 visualizes the attribute importances for the positive sample. The positively contributing attributes *excellent* and *pleasant* are identified correctly by the methods in most of the cases, *suprise* could not be detected by the Word Omission method. Negatively affecting attribute *unwatchable* is captured by explanation methods in all cases of the LSTM model. Integrated Gradients and Word Omission methods using Distill BERT model are able to detect only a certain number of decomposed sub-words, not the whole. While comparing the outcomes of the two models, it's worth considering the magnitudes of positive and negative attributes, LSTM model has higher values than DistillBERT model.

Methods	LSTM	Distill BERT
Prediction Probabilities	[0.245, 0.754]	[0.007, 0.992]
SHAP	this −0.120 was −0.073 excellent 0.612 and 0.066 a −0.0007 pleasant 0.269 surprise 0.212 since 0.044 i −0.005 have 0.005 found −0.127 most 0.043 star 0.026 wars 0.156 offerings 0.063 unwatchable −0.833 lately 0.068	this 0.013 was 0.002 excellent 0.230 and 0.019 a 0.061 pleasant 0.121 surprise 0.080 since 0.022 i 0.0017 have 0.011 found 0.0025 most 0.023 star 0.0016 wars 0.0057 offerings −0.0117 un −0.062 watch −0.0331 able −0.0361 lately −0.009
LIME	this −0.008 was 0.004 excellent 0.231 and 0.043 a 0.024 pleasant 0.175 surprise 0.159 since 0.073 i −0.009 have 0.039 found −0.096 most 0.041 star −0.005 wars 0.004 offerings 0.043 unwatchable −0.732 lately 0.103	this −0.0029 was 0.030 excellent 0.125 and 0.011 a −0.002 pleasant 0.085 surprise 0.079 since 0.039 i −0.006 have −0.032 found −0.0008 most 0.023 star 0.028 wars −0.005 offerings −0.027 unwatchable −0.035 lately −0.020
Integrated Gradients	this −0.014 was 0.047 excellent 0.278 and 0.022 a 0.23 pleasant 0.181 surprise 0.216 since 0.017 i −0.048 have 0.002 found −0.081 most −0.032 star −0.034 wars 0.014 offerings −0.006 unwatchable −0.149 lately 0.007	this 0.062 was 0.018 excellent 0.224 and 0.030 a 0.0216 pleasant 0.092 surprise 0.059 since 0.011 i 0.011 have 0.006 found 015 most 0.033 star 0.012 wars 0.003 offerings −0.015 un −0.072 watch −0.005 able −0.031 lately −0.011
Word Omission	this −0.075 was −0.027 excellent 0.680 and −0.137 a −0.170 pleasant 0.242 surprise −0.106 since −0.192 i −0.2143 have −0.201 found −0.235 most 0.23 star −0.2141 wars −0.074 offerings −0.187 unwatchable −0.245 lately 0.447	this 0.0003 was −0.00058 excellent 0.005 and 0.0012 a −0.00013 pleasant 0.0014 surprise 0.0017 since −0.00038 i 0.000001 have 0.0011 found 0.00011 most 0.0015 star 0.00010 wars 0.0000001 offerings 0.00015 un −0.00034 watch 0.00013 able −0.00053 lately −0.00010
Random	this was excellent and a pleasant surprise since i have found most star wars offerings unwatchable lately	this was excellent and a pleasant surprise since i have found most star wars offerings un watch able lately

Table 4.2.: Evaluation of explanations for simple text which is classified as positive review

As a result of comparisons involving two samples from the IMDB data set, the probability-based prediction results do not necessarily lead to higher local fidelity, implicitly the correctness property of the quantitative approach. The explanation methods do not perform well when capturing negative contributing attributes, unlike positive contributing attributes.

4.2. Automatic Evaluation Methods

In this section, the outputs of automatic evaluation methods are presented for randomly sampled 500 instances. Selected explanation methods such as Partition Explainer of SHAP, Lime with 5000 samples for the LSTM model and 500 samples for the DistillBERT model, Integrated Gradients and Word Omission techniques are evaluated on four different data sets. Noteworthy to indicate that LIME generates the attribute importances for each word separately but position information is not provided. Similarly, Word Omission estimates the feature importance for each unique word, therefore we suppose to mask all occurrences of the words, despite SHAP and Integrated Gradients being compatible with locating the position of words. Section 4.2.1 demonstrates the detailed comparison of explanation methods based on three evaluation methods and Section 4.2.2 explicates run times for LSTM and DistillBERT models. The following two subsections elucidate the evaluation of switching points and identity properties for the LSTM model.

4.2.1. Summary Statistics

The experiment design is built upon masking all occurrences of words while using the LSTM model and vocabulary indexes for the DistillBERT model, thereafter switching attribution masks of related indexes. AOPC, Log Odds and Relative AOPC metrics compare the probabilistic prediction in terms of distinct measures before and after the replacement. Along the lines of [13, 11], the aim is to examine the results based on the change thereafter masking a certain amount (called K) of attributes. The key point in the evaluation procedure is properly grouping the attribution importances, thus we have split the importance values into three groups, *positive*, *negative* and *all* according to the contribution to prediction. For the evidence groups *positive* and *negative*, the importance is sorted by the magnitude of the value taken from the explanation method, but for evidence group *all*, the importance values are sorted by ascending. K represents the number of attributes that are selected within the contribution group regarding the magnitude. For instance, $K = 2$ and evidence group *positive* means that the first two most positively contributed attributes are removed(masked) from all samples and the shift in probability is compared using evaluation metrics.

Our main focus is evaluating the explanation methods based on evidence groups for the different data sets which have different characteristics, therefore it centers on the LSTM model, later with the DistillBERT implementation, which allows us to compare the explanation methods for the models based on summary statistics. While LSTM is the base model for explanation methods, it is apparent that the changes in evaluation metrics have similar trends for contributions *all* and *positive*, hence only evidence *all* is used for comparison. Moreover, another reason for evaluating the methods for only evidence *all* is generating outcomes for the DistillBERT model takes a long time, it is elaborated in Section 4.2.2. The tables in this subsection consist of the outputs for evidence group *all* and $K = 2, 5, 10, 20$. The number of test samples is determined as 500 to have acceptable coverage and variance among the samples and the selected sample size is verified by performing the evaluation steps for 1000 random samples from the IMDB data set. The outcome shows that evaluation based on the data of 500 samples is statistically meaningful enough with having feasible computation time, associated results are observable in Table 4.3 4.5 4.6.

In the first part, AOPC scores are represented, here it simply considers before and after probabilities on predicted class, therefore when the negative contributed features are removed from the text, the prediction should reflect strength in probability. For contribution *negative*, AOPC values should be negative. On the contrary for a *positive* contribution, the preferable result is a score close to 1. In essence, *all* generated parallel results to *positive*, since all features are ordered by ascending and they were removed according to K. If the text consists of sufficient positive important features, then it ends up with the same result of *positive*, otherwise also the effects of *negative* features should be taken into consideration.

The first metric to evaluate the performance of explanation methods is Area Under Perturbation Curve(AOPC) score. It fundamentally calculates the average difference in the predicted class before and after masking the K feature on the texts. Higher values are the indicator of better performant methods. As illustrated in Figure 4.1, AOPC scores of the explanation methods based on the LSTM model's outputs, are correlated with the number of removed features for evidence group *positive* and *all*, nevertheless, the increase between 10 and 20 removed features is not notable for the Fake News and Spam data sets. The average sentence length for the Spam data set is 14 words, therefore it is possible to remove less than 20 features and that might cause no difference between 10 and 20 for K. Although the average sentence length and cut-off value for the model are around 411, still the improvement in the AOPC score between 10 and 20 removed features is not evident, since the texts are very long, the effect of tokens might be lower and the task is more complex than other data sets, hereby we can argue that when the text is too long or too short, the attribution based explanation methods might not reveal great per-

formance. A similar situation occurs for negative evidence. When the *negative* evidence is removed from texts, AOPC should be negative, since the prediction should make progress further on the predicted class, however, SHAP, LIME, and IG for Yelp and IMDB data sets show a similar trend(with very small magnitude) to evidence *positive* and *all* unexpectedly. AOPC scores of Fake News and Spam data sets evolve around 0 independent from K. The Word Omission method presents negative AOPC scores for IMDB and Yelp data sets. The fact that related to negative evidence and emerged in our experiments, was remarked in Nguyen et al.[13]'s work. The work represents the not significant AOPC values(close to zero values) for negative evidence in many cases.

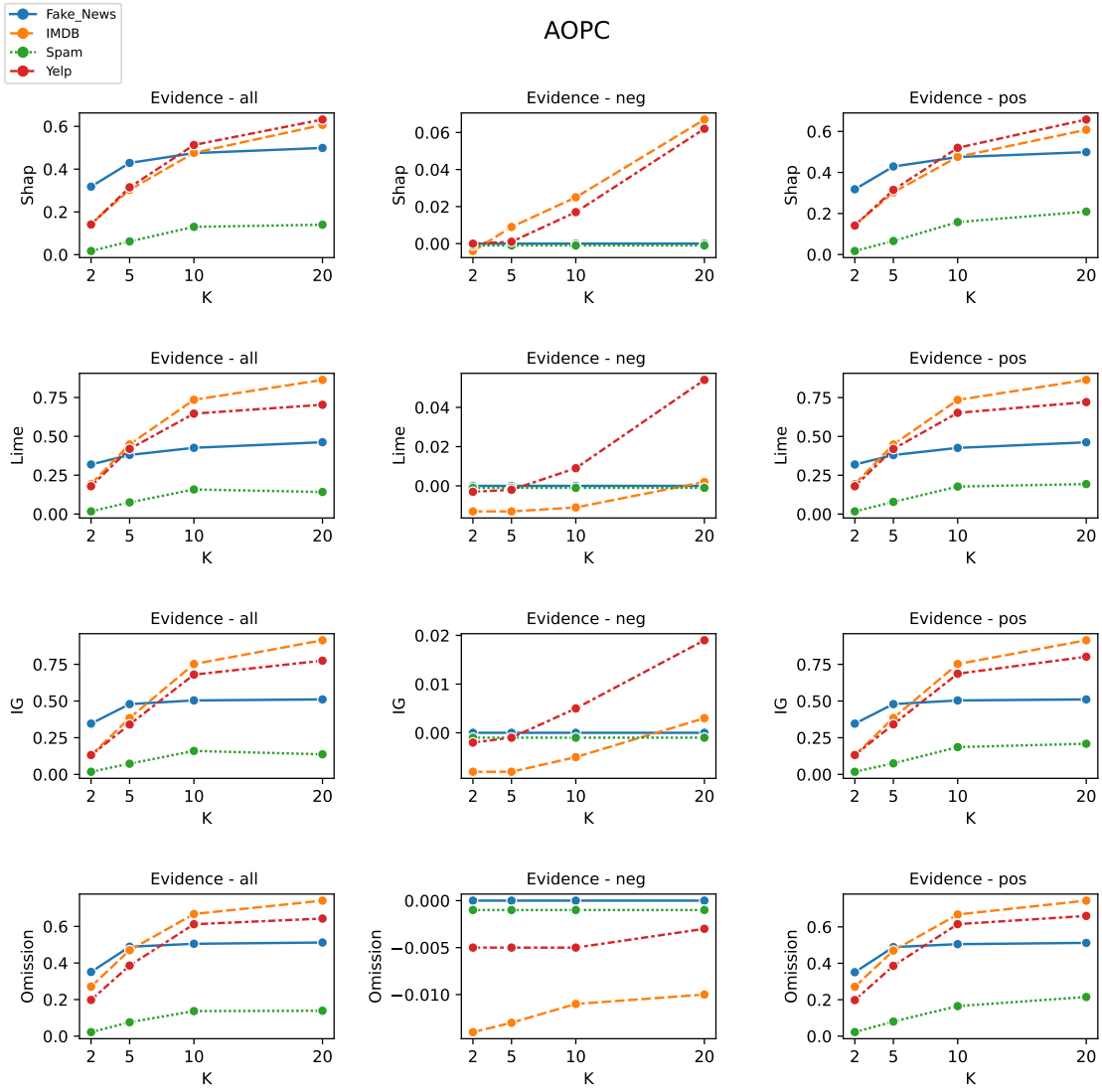

Figure 4.1.: LSTM - AOPC Score

Model	Data Set	K	SHAP	LIME	IG	Omission	Random
LSTM (500 samples)	Fake News	2	0.318	0.319	0.346	0.351	0.001
		5	0.428	0.379	0.478	0.489	0.007
		10	0.475	0.426	0.504	0.504	0.010
		20	0.498	0.461	0.511	**0.512**	0.039
	Yelp	2	0.141	0.178	0.132	0.197	0.022
		5	0.315	0.418	0.340	0.385	0.069
		10	0.513	0.656	0.679	0.611	0.131
		20	0.632	0.703	**0.775**	0.643	0.260
	IMDB	2	0.143	0.192	0.130	0.270	0.041
		5	0.302	0.448	0.383	0.469	0.090
		10	0.475	0.734	0.752	0.668	0.164
		20	0.607	0.863	**0.914**	0.740	0.254
	Spam	2	0.016	0.017	0.017	0.021	0.001
		5	0.062	0.075	0.072	0.076	0.008
		10	0.130	0.157	**0.160**	0.136	0.015
		20	0.140	0.142	0.136	0.138	0.103
LSTM (1000 samples)	IMDB	2	0.138	0.192	0.131	0.266	0.036
		5	0.294	0.444	0.378	0.465	0.091
		10	0.469	0.722	0.754	0.668	0.163
		20	0.598	0.870	**0.915**	0.746	0.257
DistillBERT (500 samples)	Fake News	2	0.400	0.363	0.452	0.414	0.506
		5	0.496	0.389	0.498	0.498	0.505
		10	0.504	0.411	0.505	0.503	0.506
		20	0.507	0.437	**0.510**	0.508	0.505
	Yelp	2	0.123	0.101	0.121	0.158	0.451
		5	0.254	0.170	0.271	0.319	0.452
		10	0.376	0.237	0.430	0.463	0.449
		20	0.490	0.319	**0.609**	0.613	0.449
	IMDB	2	0.082	0.069	0.088	0.141	0.412
		5	0.153	0.111	0.182	0.274	0.412
		10	0.233	0.152	0.301	0.422	0.412
		20	0.335	0.209	0.461	**0.578**	0.412
	Spam	2	0.041	0.035	0.035	0.044	0.156
		5	0.088	0.066	0.107	0.091	0.156
		10	0.151	0.094	0.250	0.146	0.162
		20	0.178	0.094	**0.408**	0.169	0.169

Table 4.3.: Area over perturbation curve(AOPC) results. Highlighted values indicate the best value of that metric (maximum for AOPC) for the evidence *all* within group for each data set.

Table 4.3 demonstrates the AOPC scores for evidence *all*. Integrated Gradients outperform other explanation methods for Yelp and Spam data sets in two deep-learning models. Nonetheless, the Word Omission method performs well for the Fake News data set while LSTM runs and for the IMDB data set while DistillBERT runs. The remarkable point is Random attribute selection method generates high and even in some cases very close AOPC values to the best explanation method's score within groups when DistillBERT is operated. The reason for the approximate performance of the Random selection method might rely on the fragility of the models or the weaknesses of explanation methods. [67] proposes a method to study the fragility of deep learning models by shuffling the positions of words in sentences. The results show that the pre-trained language model BERT is slightly more impacted than CNN and LSTM models, thus it denotes BERT model is reliant on semantic connections. Moreover [68] benchmarks the accuracies after deleting a certain amount of words according to the most important features from LIME and Random selection methods for FastText and BiLSTM models. In line with the outcomes of our work, LIME and Random have very close accuracies for FastText[16] after selected words are removed, the contrary to BiLSTM model, since LIME generates a decline in accuracies more than random selection for the BiLSTM model.

As Chen et al.[11] follows, the second metric Log-Odds calculates the logarithmic(natural) probabilities on predicted class before and after masking K features on the texts. Lower values indicate the strength of the explanation method since it indicates that the probability of predicted class drastically drops after masking the features compared to before masking. The log odds based on natural logarithm is not a linear function and the cardinality of change in prediction, influences the result logarithmically. Table 4.4 below is constituted for simplifying and illustrating the natural logarithm's functioning.

Pred. After Masking	**Pred. Before Masking**	**Ratio**	**Log-Odds**
0.1	0.75	0.13	-2.014
0.1	0.9	0.11	-2.19
0.01	0.99	0.01	-4.5

Table 4.4.: The simple log-odds outputs for simulating the results.

The Integrated Gradients method has the second (merely 0.001 less than Word Omission) best explanation method for Fake News, it reflects the highest Log-Odds value. However, the Fake News data set, while running the LSTM model, consists of outliers due to negative infinity that is required to be removed to estimate the average. Negative infinity is caused by the switch to the absolute same prediction in the opposite class. Therefore the correlation among the metrics might not be set up straight-forwardly for the Fake News data

set. Furthermore, other extreme values are zero log odds, which are the consequence of unchanged probability on predicted class because of the division corresponding to 1 therefore natural logarithm is equal to 0. Table 4.5 represents that Integrated Gradients reveals excellent performance for the Fake News data set in both models and for the IMDB data set on the LSTM model. In accordance with the AOPC results, Fake News and Spam data set yields close to zero values for the *negative* evidence based on log odd metrics. Log Odds for the BERT model data sets are in line with AOPC, but the methods that give the best value for the LSTM model, albeit with slight differences, vary for the Log Odds metric.

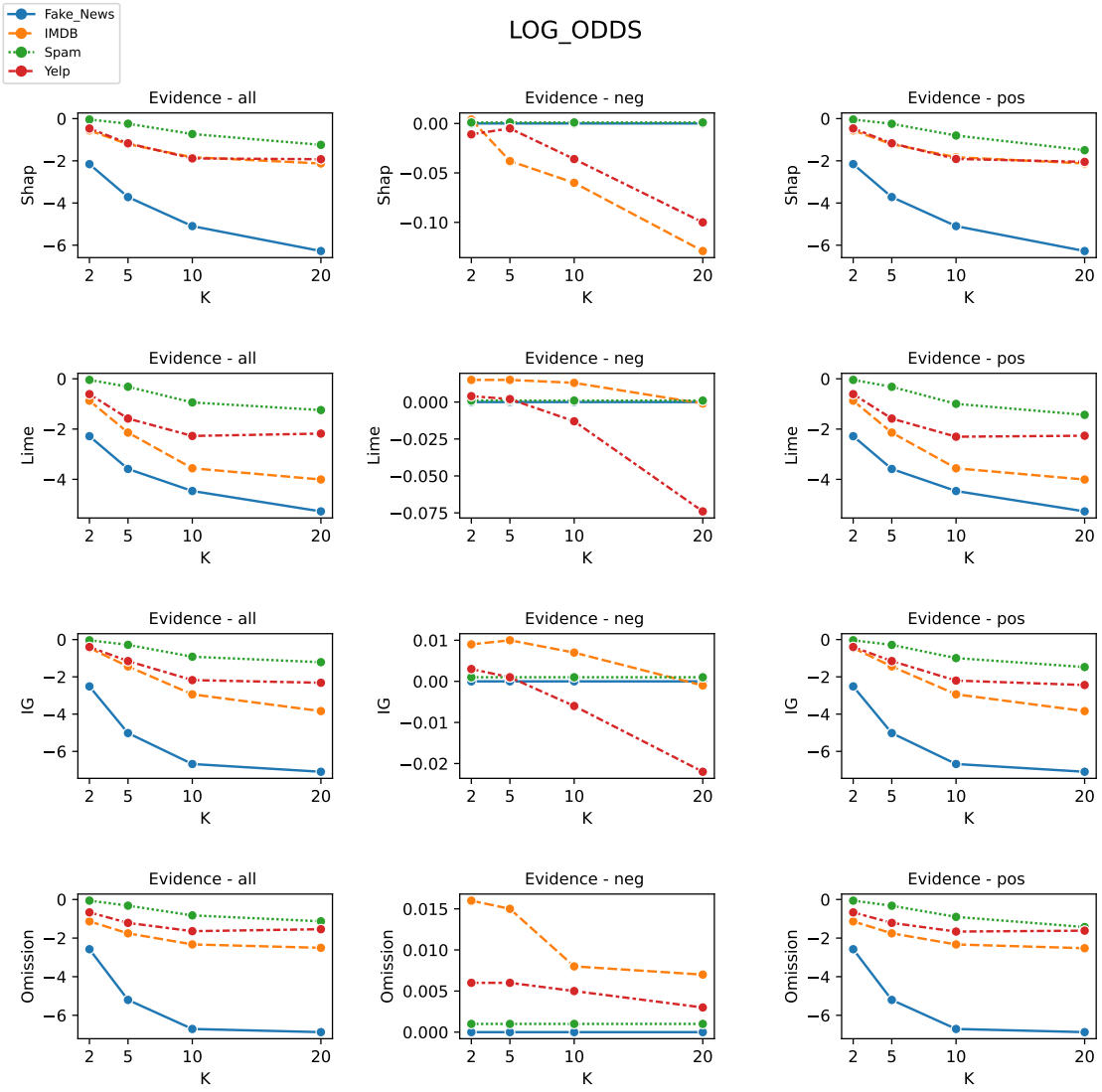

Figure 4.2.: LSTM - Log-Odds Score

Model	Data Set	K	SHAP	LIME	IG	Omission	Random
LSTM (500 samples)	Fake News	2	-2.157	-2.282	-2.514	-2.577	-0.003
		5	-3.717	-3.587	-5.017	-5.202	-0.045
		10	-5.094	-4.463	-5.017	-6.707	-0.063
		20	-6.275	-5.275	**-7.093**	-6.871	-0.2905
	Yelp	2	-0.466	-0.610	-0.399	-0.668	-0.033
		5	-1.169	-1.581	-1.155	-1.212	-0.193
		10	-1.884	-2.274	-2.174	-1.641	-0.284
		20	-1.924	-2.179	**-2.318**	-1.539	-0.564
	IMDB	2	-0.558	-0.869	-0.454	-1.133	-0.102
		5	-1.211	-2.139	-1.441	-1.753	-0.272
		10	-1.829	-3.559	-2.940	-2.328	-0.476
		20	-2.124	**-4.006**	-3.837	-2.505	-0.637
	Spam	2	-0.0423	-0.0428	-0.038	-0.0596	-0.0012
		5	-0.240	-0.315	-0.283	-0.320	-0.031
		10	-0.729	-0.939	-0.921	-0.823	-0.062
		20	-1.242	**-1.243**	-1.214	-1.129	-0.711
LSTM (1000 samples)	IMDB	2	-0.529	-0.825	-0.434	-1.085	-0.105
		5	-1.197	-2.104	-1.383	-1.744	-0.294
		10	-1.822	-3.494	-2.945	-2.348	-0.446
		20	-2.101	**-4.017**	-3.838	-2.551	-0.643
DistillBERT (500 samples)	Fake News	2	-1.302	-0.926	-1.677	-1.461	-1.675
		5	-2.258	-1.095	-2.270	-2.302	-1.672
		10	-2.579	-1.277	-2.601	-2.538	-1.678
		20	-2.845	-1.532	**-2.892**	-2.741	-1.670
	Yelp	2	-0.256	-0.226	-0.249	-0.309	-0.650
		5	-0.578	-0.383	-0.610	-0.756	-0.649
		10	-0.961	-0.575	-1.117	-1.281	-0.643
		20	-1.355	-0.831	**-1.894**	-1.871	-0.646
	IMDB	2	-0.137	-0.116	-0.145	-0.247	-0.596
		5	-0.278	-0.204	-0.331	-0.555	-0.595
		10	-0.458	-0.307	-0.619	-0.957	-0.595
		20	-0.719	-0.433	-1.047	**-1.508**	-0.596
	Spam	2	-0.080	-0.071	-0.069	-0.085	-0.564
		5	-0.185	-0.138	-0.234	-0.182	-0.562
		10	-0.406	-0.235	-0.546	-0.376	-0.571
		20	-0.681	-0.321	**-1.084**	-0.629	-0.571

Table 4.5.: Log-Odds scores. Highlighted values indicate the best value of that metric (minimum for log-odds) for the evidence *all* within the data set group.

The relative AOPC is essentially the AOPC score multiplied by the ratio of the absolute sum of the K-selected feature importance to the absolute sum of all feature importance. When we compare the explanation methods for one individual instance, if the explanation methods pick up the same features, then the AOPC metric gives the same values. Consequently, the root cause of proposing this method is to provide a foundation for the feature importance given by the explanation methods. The relative AOPC metric does not estimate the scores for Random Selection, as described in the AOPC part, Random selection does not provide feature importance, thus the rAOPC values for explanation methods are not represented in this section. Table 4.6 represents that Word Omission performed as the best method in all cases except for the Spam dataset when running the DistillBERT model. The output of Word Omission correlates directly with the prediction change on the predicted class, thereby when the metric normalized the magnitudes of features by summing them up, it increases the degree of influence in the result of the metric.

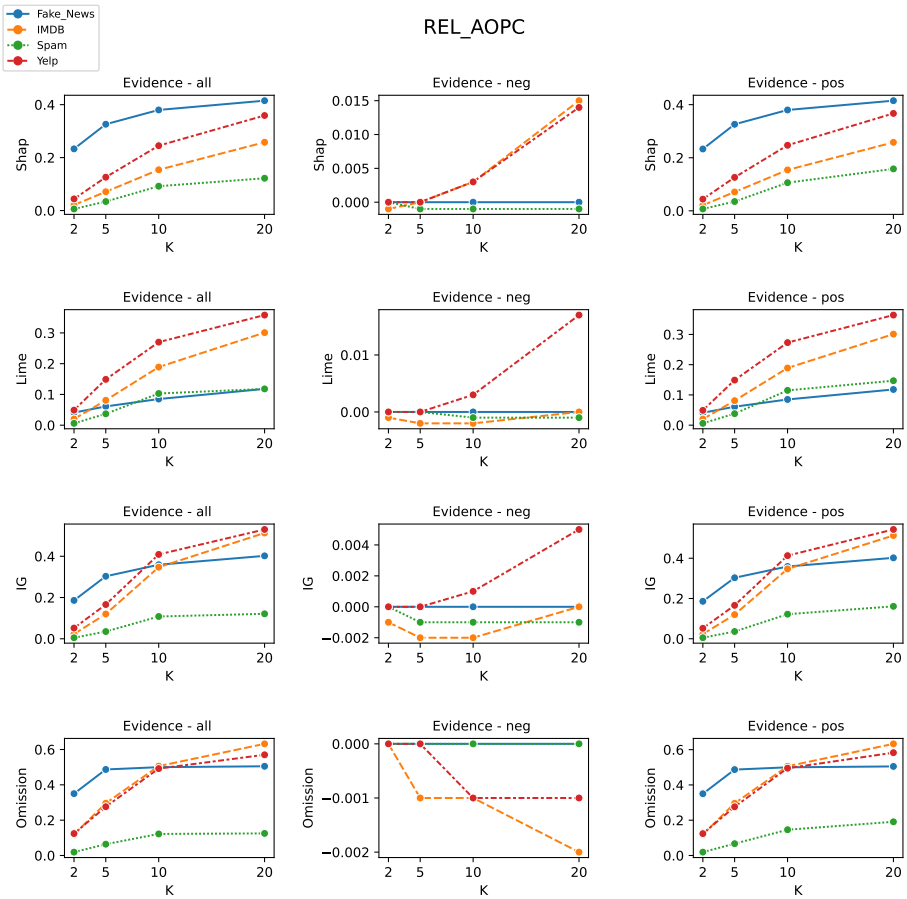

Figure 4.3.: LSTM - Relative AOPC Score

Model	Data Set	K	SHAP	LIME	IG	Omission
LSTM (500 samples)	Fake News	2	0.233	0.041	0.185	0.349
		5	0.326	0.061	0.302	0.487
		10	0.379	0.084	0.358	0.500
		20	0.414	0.117	0.402	**0.505**
	Yelp	2	0.043	0.048	0.052	0.123
		5	0.126	0.149	0.165	0.276
		10	0.244	0.270	0.409	0.492
		20	0.359	0.357	0.529	**0.569**
	IMDB	2	0.020	0.019	0.023	0.119
		5	0.070	0.080	0.119	0.295
		10	0.153	0.188	0.346	0.506
		20	0.257	0.300	0.512	**0.632**
	Spam	2	0.006	0.005	0.004	0.018
		5	0.033	0.037	0.034	0.064
		10	0.091	0.103	0.108	0.121
		20	0.122	0.117	0.120	**0.124**
LSTM (1000 samples)	IMDB	2	0.020	0.019	0.023	0.118
		5	0.069	0.080	0.118	0.298
		10	0.151	0.189	0.346	0.510
		20	0.254	0.306	0.514	**0.638**
DistillBERT (500 samples)	Fake News	2	0.353	0.034	0.287	0.411
		5	0.464	0.052	0.365	0.495
		10	0.479	0.078	0.382	0.500
		20	0.486	0.118	0.395	**0.504**
	Yelp	2	0.038	0.020	0.028	0.061
		5	0.113	0.051	0.095	0.159
		10	0.208	0.095	0.196	0.267
		20	0.324	0.161	0.345	**0.400**
	IMDB	2	0.016	0.005	0.011	0.003
		5	0.047	0.017	0.038	0.090
		10	0.098	0.036	0.092	0.175
		20	0.180	0.076	0.194	**0.294**
	Spam	2	0.015	0.011	0.010	0.022
		5	0.046	0.033	0.058	0.053
		10	0.101	0.062	0.196	0.095
		20	0.149	0.079	**0.374**	0.127

Table 4.6.: Relative area over perturbation curve(rAOPC) results. Highlighted values indicate the best value of that metric (maximum for rAOPC) for the evidence *all* within the data set group.

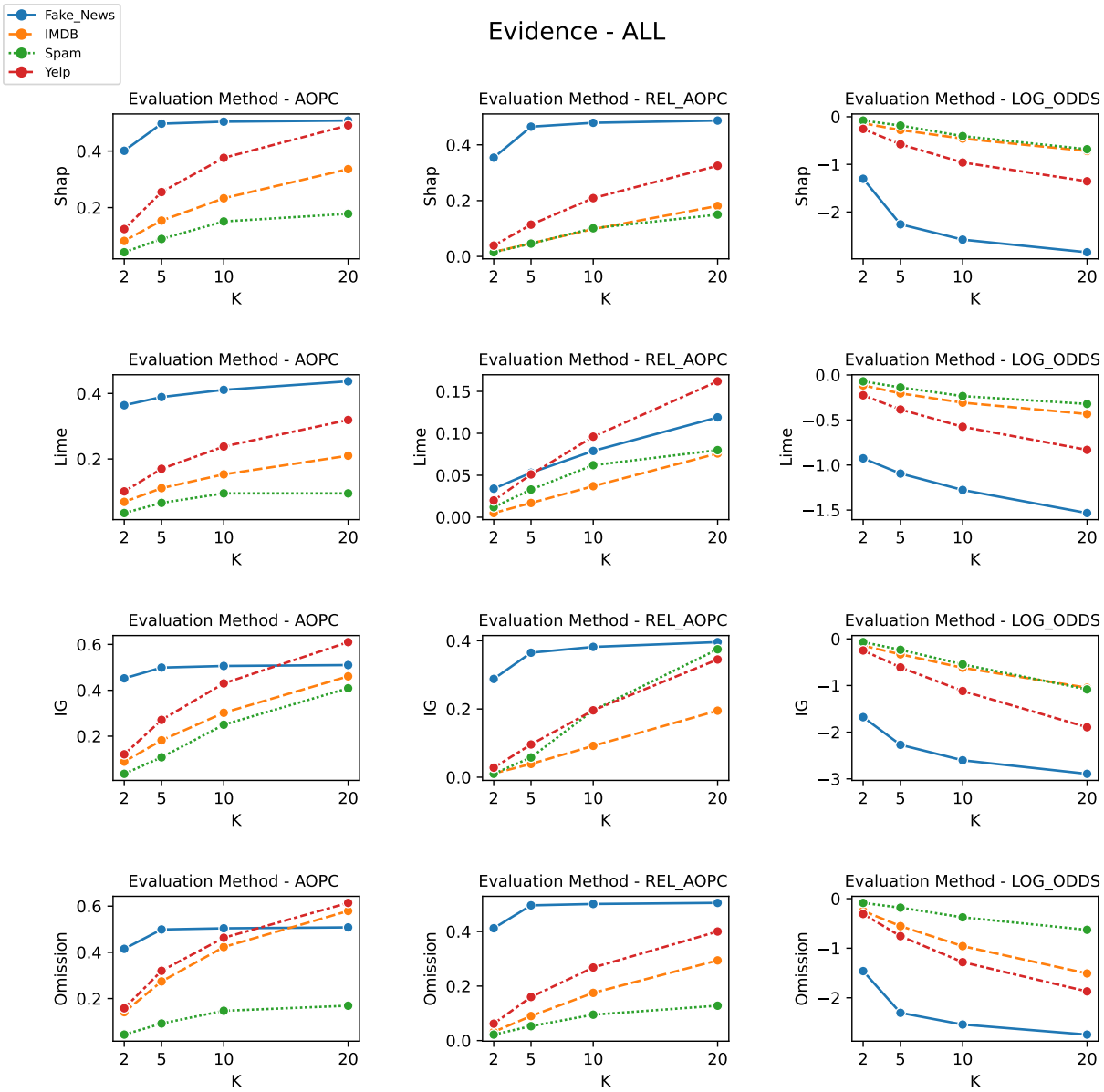

Figure 4.4.: BERT - Evaluation Scores for Evidence *all*

4.2.2. Processing Time

In this section, we represent the processing times of explanation methods for each data set and deep learning model. Table 4.7 shows that omission has obviously a big portion of time among the data sets while the LSTM model is utilized, due to its nature of finding attribute importances by removing each individual token from the sequence and estimating the change in prediction probability. Conversely, SHAP has the longest processing times for all data sets when the BERT model is used. Among the data sets, the Fake News data set has longer processing times regarding the LSTM model than other data sets, the main reason is Fake News data set has the longest sentence length parameter that affects the processing time. Nevertheless, the processing takes a long time for three explanation methods out of four methods as the BERT model is employed. Associatively with the maximum length hyper-parameter, the processing time increases or decreases as well.

Evaluation Method	The Model	Fake News	IMDB	Spam	Yelp
SHAP	LSTM	4.69	4.10	1.93	4.00
	BERT	99.03	170.37	34.44	111.95
LIME	LSTM	3.72	2.37	1.08	2.18
	BERT	0.07	0.07	0.01	0.05
IG	LSTM	4.22	3.01	1.56	6.74
	BERT	0.15	0.31	0.18	0.18
Omission	LSTM	18.03	10.28	0.85	5.76
	BERT	9.82	5.38	1.14	7.73
Random	LSTM	0.04	0.04	0.05	0.04
	BERT	0.004	0.01	0.004	0.007

Table 4.7.: The average processing times in seconds for individual samples

4.2.3. Identity Property Evaluation

As discussed in Chapter 2, the Identity property is a measure to evaluate the stability and correctness of the explanation method. It requires identical feature importances independently from running at different times but for the same sample on the same machine with the same seed. We designed an experiment that compares the feature importances after successive two runs whether they are exactly the same or not. It generates percentage-based results for each sample.

The table 4.8 represents the average percentage of the samples that do not satisfy identity property. The main reason for not ensuring the property is well known due to randomness nature of explanation methods, particularly LIME randomly samples from the neighborhood of the words to explain, this inference is also demonstrated in the study [12] using tabular data. In the same manner,

the Partition Explainer of SHAP generates various partitions in each run to measure the marginal contribution of words. Nevertheless, Integrated Gradients and Word Omission techniques do not contain any randomness throughout the process. The results demonstrate the effect of randomness in the processes. While Integrated Gradients and Word Omission methods satisfy the identity property for all samples, LIME is not able to satisfy even for one sample and SHAP has some exceptions.

Data Set	**SHAP**	**LIME**	**IG**	**Omission**
Fake News	11.68%	100%	0%	0%
Yelp	0.011%	100%	0%	0%
IMDB	0%	100%	0%	0%
Spam	0.26%	100%	0%	0%

Table 4.8.: The percentages of not satisfied identity property samples

4.2.4. Switching Point

In an effort to evaluate the performance of exploiting important features of explanation models, we have designed an experimental setup using the LSTM model. The feature importances for explanation methods respectively for all data sets are generated and iteratively most important N features are selected and removed from the text and the change in class is inspected and the switching point is determined. We have used all features in this setup independently from their positive or negative contribution. While masking all words due to reaching the maximum length of the sentence, the predictions for all data sets end up with class 0, therefore if the result of the original text is class 0, it might still end up with class 0.

Figure 4.5 shows Spam data set for all explanation methods gives the worst results because sentence length is very short and turns out to be fully masked with zero and generates class 0 results. In the Figure, the Word Omission method demonstrates that most of the samples have 100% of tokens that should be masked to switch the class, however, this doesn't reflect the truth, since these test samples originally belonged to class 0 and when we masked all tokens their switched classes are still 0. Thus when we discard 392 outliers, the average is 39.2% for 108 samples. IMDB data set has the lowest percentage of switching points among the data sets and correlatively Yelp data set has very low switching point values.

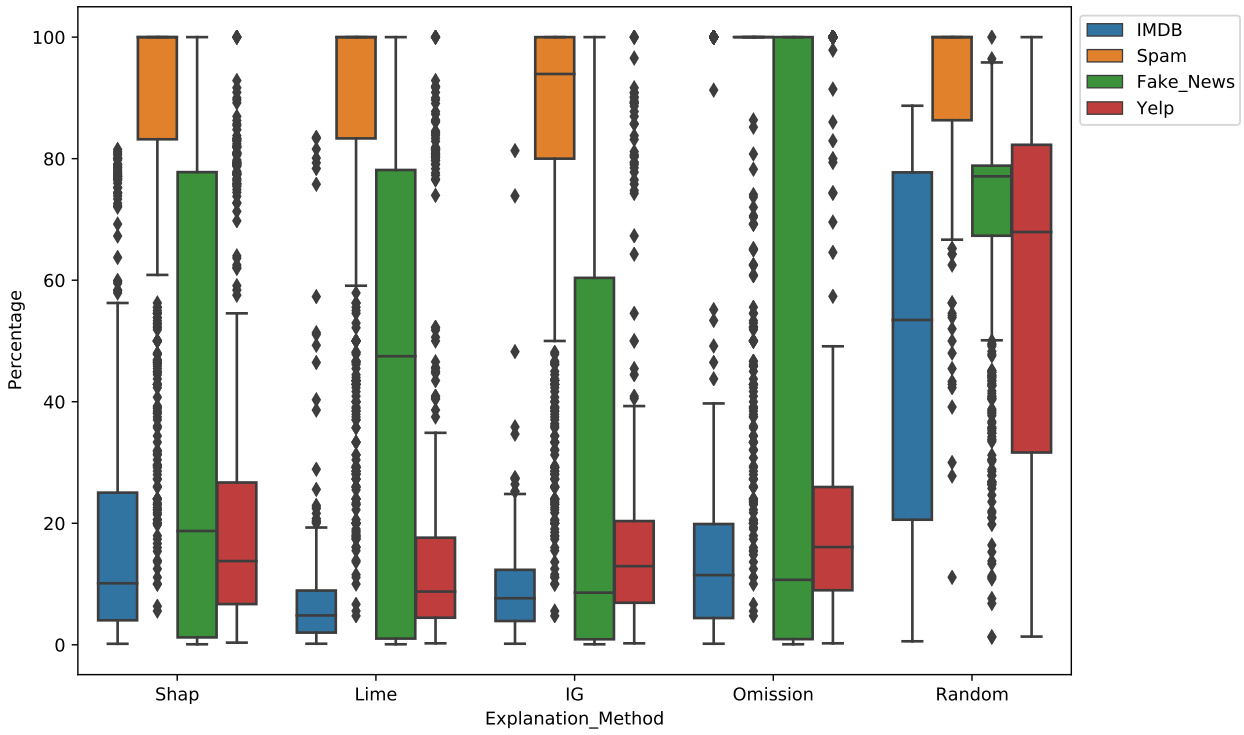

Figure 4.5.: The switching points based on explanation methods for data sets

5. Conclusion and Future Work

5.1. Conclusion

This thesis aimed to evaluate prevalent attribution-based explanation methods on black-box models. In line with this objective, previously proposed metrics in the literature are discussed thoroughly and applied to several data sets. Furthermore, a novel derived metric with some modifications is presented, capable of providing comprehension on instance-level benchmarks. Throughout experiments proved that tackling the topic from various aspects revealed the strengths and weaknesses of the evaluation methods as well.

The foundation takes the basis on two components, interpretation of the generated explanation for simple texts and automatic evaluations for batch samples. While the first part sheds light on explanation outcomes by visualizing the first three positively and negatively contributed attribute importances, the second part serves to depict the whole picture. In the first part, the outcomes show that explanation methods may claim similar attributes are essential for the decision; nonetheless, their weights may not be associative. On top of that, a high probability of predicted class does not necessarily stimulate a superior result. The model performance varies using different model architectures. This situation recalls the ground of proposing the modified novel metric to extract a better performant method substantially.

The summary statistics tables and graphs show that the increase in the number of masked words leads to a positive increase in all 3 metrics for AOPC, both positively and when all features are taken into account. Nevertheless, the removal of words with a high negative impact has a reverse effect with the increase in the number of words, and the effect is even larger for all methods when more than 10 words are removed. This may be due to the fact that words with negative significance do not actually have a negative impact beyond the 10th rank when ranked in absolute order of magnitude. Log-Odds and Relative Odds methods represent correlative results to AOPC with small differences in the best performant method. Hence the situation brings on the interrogation of the success of attribution-based explanation methods.

Moreover, the other metrics, identity property, switching point and processing times, enabled the methods to be considered from different aspects. When the processing times of the Explanation methods are compared for the two models, it is apparent that Word Omission and SHAP give inversely proportional results for these models, and the maximum number of words affects the

processing time. Identity Property shows the effect of randomness in the explanation methods since the outcomes could be different in successive runs. Switching Point represents the percentage of necessary words to be masked in the sentence to the outcome to be turned to the opposite class. According to the results, Integrated Gradients outperform other methods since it has smaller bound ranges for data sets than other data sets, which means it is possible to turn to the prediction of other classes with the masked words found by integrated Gradients.

Explanation methods and the black-box models possess weaknesses. In our configured setup, only word-based attribution importances are considered; however, phrase-based and subject-verb-object-based relations carry the meaning of the sentences, a.k.a semantic relation. Therefore they need to be taken into account in a particular approach. Furthermore, we have noticed that some explanation methods, such as LIME, need to point out the attribute's position and that it is impossible to figure out exactly which word was affected if there is more than one repetition of the same word. Besides, as we mentioned in Section 4.2.1, the deep learning models might be vulnerable and easily fooled by even randomly selected features.

5.2. Future Work

In this section, we aim to focus on future work in consideration of related work and our study. The study was mainly based on extensively used explanation methods and open-source data sets which consist of binary classified data. As an extension of this work, the evaluation procedure might be applied to multi-class classification as well as using more fine-grained progressive local explanation approaches such as XPROAX (Local Explainer with Progressive Neighborhood Approximation)[69] the methods using neighborhood words around the instance word to explain in local boundaries, HEDGE (Hierarchical Explanation via Divisive Generation) [11] that provided hierarchical explanations for phrases. Since even the methods outperform popular explanation methods, still require to be evaluated in terms of processing time, the flexibility of choosing a number of units, ease of configuration, and switching units by percentage. In addition to specific explanation methods, selected important features might be analyzed in terms of part of speech tagging, particularly distribution among the share of important features and the correlation with the local fidelity.

Bibliography

[1] Ethics guidelines for trustworthy ai. `https://ec.europa.eu/futurium/en/ai-alliance-consultation.1.html`. Accessed: 2022-11-16.

[2] Complete guide to gdpr compliance. `https://www.gdpr.eu/tag/gdpr/`. Accessed: 2022-11-16.

[3] Finale Doshi-Velez and Been Kim. Considerations for evaluation and generalization in interpretable machine learning. 2018.

[4] Gabrielle Ras, Marcel van Gerven, and Pim Haselager. Explanation methods in deep learning: Users, values, concerns and challenges, 2018.

[5] Leon Sixt, Maximilian Granz, and Tim Landgraf. When explanations lie: Why many modified bp attributions fail, 2019.

[6] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods, 2019.

[7] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[8] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

[9] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

[10] Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Explaining predictions of non-linear classifiers in nlp, 2016.

[11] Hanjie Chen, Guangtao Zheng, and Yangfeng Ji. Generating hierarchical explanations on text classification via feature interaction detection, 2020.

[12] Milo Honegger. Shedding light on black box machine learning algorithms: Development of an axiomatic framework to assess the quality of methods that explain individual predictions, 2018.

[13] Dong Nguyen. Comparing automatic and human evaluation of local explanations for text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages

1069–1078, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[14] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[16] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information, 2016.

[17] Andrea Gasparetto, Matteo Marcuzzo, Alessandro Zangari, and Andrea Albarelli. A survey on text classification algorithms: From text to predictions. *Information*, 13(2), 2022.

[18] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China, July 2015. Association for Computational Linguistics.

[19] Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. Automatic labelling of topics with neural embeddings. 2016.

[20] Lei Xiang, Qian Wang, Xiyao Jin, Dong Nie, Yu Qiao, and Dinggang Shen. Deep embedding convolutional neural network for synthesizing ct image from t1-weighted mr image, 2017.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[22] Rie Johnson and Tong Zhang. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 562–570, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[23] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[24] Hannah Kim and Young-Seob Jeong. Sentiment classification using convolutional neural networks. *Applied Sciences (Switzerland)*, 9, 06 2019.

[25] Andor Diera, Bao Xin Lin, Bhakti Khera, Tim Meuser, Tushar Singhal, Lukas Galke, and Ansgar Scherp. Bag-of-words vs. sequence vs. graph vs. hierarchy for single- and multi-label text classification, 2022.

[26] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[27] Ozan Irsoy and Claire Cardie. Deep recursive neural networks for compositionality in language. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2096–2104, Cambridge, MA, USA, 2014. MIT Press.

[28] Moses Soh. Learning cnn lstm architectures for image caption generation. 2016.

[29] Adil Moghar and Mhamed Hamiche. Stock market prediction using lstm recurrent neural network. *Procedia Computer Science*, 170:1168–1173, 2020. The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

[30] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis C. M. Lau. A c-lstm neural network for text classification, 2015.

[31] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[33] Andrea Galassi, Marco Lippi, and Paolo Torroni. Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10):4291–4308, oct 2021.

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[35] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[36] Zachary C. Lipton. The mythos of model interpretability, 2016.

[37] Cynthia Rudin and Joanna Radin. Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From an Explainable AI Competition. *Harvard Data Science Review*, 1(2), nov 22 2019. https://hdsr.mitpress.mit.edu/pub/f9kuryi8.

[38] Wei Zhao, Rahul Singh, Tarun Joshi, Agus Sudjianto, and Vijayan N. Nair. Self-interpretable convolutional neural networks for text classification. `https://arxiv.org/abs/2105.08589`, 2021.

[39] David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks, 2018.

[40] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Bach, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned, 2015.

[41] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8), 2019.

[42] Christoph Molnar. *Interpretable Machine Learning*. 2019.

[43] M. Robnik-Sikonja and Marko Bohanec. Perturbation-based explanations of prediction models. In *Human and Machine Learning*, 2018.

[44] Lime github page. `https://github.com/marcotcr/lime`. Accessed: 2022-11-16.

[45] L. S. Shapley. *A Value for n-Person Games*, pages 307–318. Princeton University Press, Princeton, 2016.

[46] Shap's partition explainer for language models. `https://towardsdatascience.com/shaps-partition-explainer-for-language-models-ec2e7a6c1b77`. Accessed: 2022-08-27.

[47] Shap github page. `https://github.com/slundberg/shap`. Accessed: 2022-08-27.

[48] José Giménez and Albina Puente. A new procedure to calculate the owen value. pages 228–233, 01 2017.

[49] G. Frederick Owen. Values of games with a priori unions. 1977.

[50] Integrated gradients documentation. `https://docs.seldon.io/projects/alibi/en/stable/methods/IntegratedGradients.html`. Accessed: 2022-11-16.

[51] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks, 2013.

[52] Martin Thoma. Analysis and optimization of convolutional neural network architectures. Masters's thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, June 2017.

[53] Marko Robnik-Šikonja and Igor Kononenko. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):589–600, 2008.

[54] Shixia Liu, Xiting Wang, Mengchen Liu, and Jun Zhu. Towards better analysis of machine learning models: A visual analytics perspective, 2017.

[55] Amin Nayebi, Sindhu Tipirneni, Brandon Foreman, Chandan K. Reddy, and Vignesh Subbian. An empirical comparison of explainable artificial intelligence methods for clinical data: A case study on traumatic brain injury, 2022.

[56] Shap documentation bar chart visualization. `https://shap.readthedocs.io/en/latest/example_notebooks/api_examples/plots/bar.html`. Accessed: 2022-11-16.

[57] Wilson Silva, Kelwin Fernandes, Maria Cardoso, and Jaime Cardoso. *Towards Complementary Explanations Using Deep Neural Networks: First International Workshops, MLCN 2018, DLF 2018, and iMIMIC 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16-20, 2018, Proceedings*, pages 133–140. 09 2018.

[58] Tiago A. Almeida, José María G. Hidalgo, and Akebo Yamakami. Contributions to the study of sms spam filtering: New collection and results. In *Proceedings of the 11th ACM Symposium on Document Engineering*, DocEng '11, page 259–262, New York, NY, USA, 2011. Association for Computing Machinery.

[59] Uci sms spam collection data set. `https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection`. Accessed: 2022-11-16.

[60] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[61] Imdb dataset of 50k movie reviews. `https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews`. Accessed: 2022-08-20.

[62] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification, 2015.

[63] Yelp review polarity data set. `https://www.kaggle.com/datasets/irustandi/yelp-review-polarity`. Accessed: 2022-11-16.

[64] Hadeer Ahmed, Issa Traoré, and Sherif Saad. Detection of online fake news using n-gram analysis and machine learning techniques. In *ISDDC*, 2017.

[65] Kaggle fake and real news dataset. `https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset`. Accessed: 2022-11-16.

[66] Hugging face - distillbert implementation. `https://huggingface.co/docs/transformers/model_doc/distilbert`. Accessed: 2022-11-16.

[67] Rutuja Taware, Shraddha Varat, Gaurav Salunke, Chaitanya Gawande, Geetanjali Kale, Rahul Khengare, and Raviraj Joshi. ShufText: A simple black box approach to evaluate the fragility of text classification models. In *Machine Learning, Optimization, and Data Science*, pages 235–249. Springer International Publishing, 2022.

[68] Utkarsh Desai, Srikanth Tamilselvam, Jassimran Kaur, Senthil Mani, and Shreya Khare. Benchmarking popular classification models' robustness to random and targeted corruptions, 2020.

[69] Yi Cai, Arthur Zimek, and Eirini Ntoutsi. XPROAX-local explanations for text classification with progressive neighborhood approximation. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, oct 2021.

Appendix

A. Algorithms

Algorithm 1 Word Omission

Require: Document D, predict_proba ▷ predict_proba gives the probabilities for each class

Ensure: $I[\,]$ ▷ the array of feature importances

function FEATURE_IMPORTANCES(D)
  $x \leftarrow transform(D)$
  $[w_0, w_1, ..., w_N] \leftarrow x$ ▷ x represents the sequences
  $p(y|x) \leftarrow predict\_proba(x)$
  for $i \leftarrow 1$ to N **do**
    $p(\hat{y}|x_{\setminus w_i}) \leftarrow predict\_proba(x_{\setminus w_i})$
    $I \leftarrow append(p(y|x) - p(\hat{y}|x_{\setminus w_i}))$
  end for
  return I
end function

Algorithm 2 Switching point

function SWITCHING_POINT($DocumentD$)
  $x \leftarrow transform(D)$,$count \leftarrow 0$
  $[w_0, w_1, ..., w_N] \leftarrow x$ ▷ x represents the sequences
  $sorted\_list \leftarrow sort\_descending([w_0, w_1, ..., w_N])$
  $y^{0,1} \leftarrow predict\_class(x)$
  for $i \leftarrow 1$ to N **do**
    $y^{'} \leftarrow predict\_class(x_{\setminus w_{1..i}})$
    if $y \neq y^{'}$ **then return** $count/N$
    else
      $count \leftarrow count + 1$
    end if
  end for
end function

B. The Interpretation of Explanation Outcomes for Other Data Sets

B.1. Yelp Reviews

Methods	LSTM	Distill BERT
Prediction Probs.	[0.9999, 0.0001]	[0.947 0.052]
SHAP	*first* −0.04540 · *german* 0.26168 · *beerhouse* −0.00818 · *experience* 0.11248 · *and* 0.000822 · *the* 0.02450 · *food* −0.0568 · *was* −0.04939 · *great* 0.27425 · *not* 0.01858 · *disappointed* −0.4613 · *great* 0.1792 · *prices* −0.0633	*first* 0.065126 · *german* 0.01348 · *beer* 0.020094 · *house* 0.01642 · *experience* 0.043022 · *and* 0.053040 · *the* −0.01941 · *food* 0.007330 · *was* −0.04367 · *great* 0.17381 · *not* 0.22717 · *disappointed* −0.19659 · *great* 0.093883 · *prices* −0.030305
LIME	*first* 0.0563 · *german* 0.21594 · *beerhouse* −0.00150 · *experience* 0.03070 · *and* 0.06334 · *the* 0.05333 · *food* −0.0432 · *was* −0.03280 · *great* 0.27262 · *not* 0.22876 · *disappointed* −0.23072 · *great* 0.13275 · *prices* −0.02438	*first* 0.085115 · *german* −0.051485 · *beerhouse* −0.000912 · *experience* 0.047474 · *and* 0.08642 · *the* 0.048415 · *food* 0.04237 · *was* −0.10908 · *great* 0.104381 · *not* 0.38658 · *disappointed* −0.18847 · *great* 0.16672 · *prices* −0.06729
Integrated Gradients	*first* 0.07501 · *german* 0.3550 · *beerhouse* −0.0500 · *experience* 0.05158 · *and* −0.04682 · *the* −0.02630 · *food* −0.1091 · *was* −0.10996 · *great* 0.18969 · *not* 0.00570 · *disappointed* −0.21385 · *great* 0.08458 · *prices* −0.01855	*first* 0.04964 · *german* 0.04683 · *beer* 0.00377 · *house* 0.021062 · *experience* 0.03833 · *and* 0.02365 · *the* 0.00019 · *food* −0.0762 · *was* −0.04559 · *great* 0.29377 · *not* −0.1190 · *disappointed* 0.06093 · *great* 0.118724 · *prices* −0.00127
Word Omission	*first* 0.19422 · *german* 0.36638 · *beerhouse* −0.00609 · *experience* 0.045651 · *and* 0.010658 · *the* −0.01514 · *food* −0.03539 · *was* −0.03240 · *great* 0.39914 · *not* 0.778207 · *disappointed* 0.048234 · *great* 0.39914 · *prices* −0.01074	*first* 0.0086 · *german* −0.00054 · *beer* 0.00227 · *house* −0.00074 · *experience* 0.0100 · *and* 0.01653 · *the* 0.00061 · *food* −0.00288 · *was* −0.0101 · *great* 0.07572 · *not* 0.83150 · *disappointed* 0.86411 · *great* 0.00321 · *prices* −0.00647
Random	*first* *german* *beerhouse* *experience* *and* *the* *food* *was* *great* *not* *disappointed* *great* *prices*	*first* *german* *beer* *house* *experience* *and* *the* *food* *was* *great* *not* *disappointed* *great* *prices*

Table .1.: Evaluation of explanations for simple text which is classified as positive Yelp review. It is taken from a restaurant's reviews on Yelp website on the date of 16/11/2022.

Methods	LSTM	Distill BERT
Prediction Probs.	[0.999995, 0.000005]	[0.9984, 0.00150]
SHAP	*worst* 0.169801 · *place* −0.020892 · *ever* 0.034135 · *all* −0.00350 *the* −0.00755 · *servers* 0.00654 · *are* −0.003983 · *rude* 0.19618 *and* −0.01320 · *the* −0.012629 · *food* −0.009400 · *is* −0.01113 *really* −0.02872 · *bad* 0.171559 · *i* −0.0171258 · *would* 0.01959 *not* 0.282963 · *recommend* −0.06282 · *anyone* 0.024589	*worst* 0.125826 · *place* 0.00899 · *ever* 0.010707 · *all* 0.008176 · *the* 0.012923 *servers* 0.022914 · *are* 0.0240757 · *rude* 0.0641111 · *and* −0.004248 *the* 0.007119 · *food* 0.031081 · *is* 0.006901 · *really* 0.02020 · *bad* 0.104262 *i* −0.003730 · *would* 0.021408 · *not* 0.098796 · *recommend* −0.024651 *anyone* 0.02212
LIME	*worst* −0.04420 · *place* 0.0032930 · *ever* 0.001851 · *all* −0.00042 *the* −0.000918 · *servers* −0.0020 · *are* 0.0031133 · *rude* −0.03428 *and* −0.002843 · *the* 0.000395 · *food* −0.00639 · *is* −0.003449 *really* −0.00200 · *bad* −0.025334 · *i* 0.004675 · *would* 0.003168 *not* −0.03684 · *recommend* 0.034669 · *anyone* 0.001009	*worst* 0.051654 · *place* 0.012583 · *ever* 0.000120 · *all* 0.000104 *the* −0.00690 · *servers* 0.005928 · *are* · *rude* 0.017625 · *and* −0.0072 *the* −0.00690 · *food* 0.019981 · *is* 0.020485 · *really* −0.00228 *bad* 0.025702 · *i* 0.001634 · *would* 0.021791 · *not* 0.02505 *recommend* −0.01829 · *anyone* 0.004686
Integrated Gradients	*worst* 0.550915 · *place* 0.015103 · *ever* 0.032054 · *all* 0.018343 *the* −0.002411 · *servers* 0.055496 · *are* 0.010657 · *rude* 0.005497 *and* 0.011448 · *the* 0.00091 · *food* 0.00774 · *is* 0.0024784 *really* 0.00299 · *bad* −0.001364 · *i* 0.001869 · *would* 0.001040 *not* −0.00148 · *recommend* 0.00289 · *anyone* 0.000125	*worst* 0.06251 · *place* 0.003945 · *ever* 0.010559 · *all* 0.00982 · *the* 0.015099 *servers* 0.008754 · *are* 0.027481 · *rude* 0.07110 · *and* 0.02820 · *the* 0.005130 *food* 0.02608 · *is* 0.029466 · *really* 0.03605 · *bad* 0.1322 · *i* −0.00092 *would* 0.03758 · *not* 0.05011 · *recommend* 0.00905 · *anyone* 0.008847
Word Omission	*worst* 0.00091438 · *place* $16.517e^{-10}$ · *ever* $66.2e^{-10}$ · *all* $7.97e^{-10}$ *the* $7.24e^{-10}$ · *servers* $3.68e^{-10}$ · *are* $1.44e^{-10}$ · *rude* $33.6e^{-10}$ *and* $6.14e^{-10}$ · *the* $7.24e^{-10}$ · *food* $16.5e^{-10}$ · *is* $2.15e^{-10}$ *really* $14.5e^{-10}$ · *bad* $21.3e^{-10}$ · *i* $8.23e^{-10}$ · *would* $14.7e^{-10}$ *not* $106.9e^{-10}$ · *recommend* $42.3e^{-10}$ · *anyone* $4.09e^{-10}$	*worst* $-3.325e^{-5}$ · *place* $-1.627e^{-5}$ · *ever* $4.458e^{-5}$ · *all* $1.209e^{-5}$ *the* 0.0 · *servers* $3.397e^{-5}$ · *are* $1.901e^{-5}$ · *rude* $3.468e^{-5}$ · *and* $0.971e^{-5}$ *the* $-1.114e^{-5}$ · *food* $4.845e^{-5}$ · *is* $-7.152e^{-7}$ · *really* $1.114e^{-5}$ *bad* $-1.198e^{-5}$ · *i* $0.441e^{-6}$ · *would* $1.162e^{-5}$ · *not* 0.00018632412 *recommend* $3.969e^{-5}$ · *anyone* $1.341e^{-5}$
Random	*worst place ever all the servers* *are rude and the food is really* *bad i would not recommend anyone*	*worst place ever all the servers* *are rude and the food is really* *bad i would not recommend anyone*

Table .2.: Evaluation of explanations for simple text which is classified as negative Yelp review. It is taken from a restaurant's reviews on Yelp website on the date of 16/11/2022.

B.2. SMS Spam

Methods	LSTM	Distill BERT
Prediction Probs.	[0.9998, 0.00016]	[0.0177, 0.9822]
SHAP	*sms* 0.119 · *auction* 0.123 · *you* 0.068 · *have* 0.148 · *won* 0.237 · *a* 0.042 · *nokia* 0.260	*sms* 0.320 · *auction* 0.311 · *you* −0.120 · *have* 0.006 · *won* 0.243 · *a* 0.043 · *nokia* 0.172
LIME	*sms* 0.177 · *auction* 0.213 · *you* 0.064 · *have* 0.164 · *won* 0.396 · *a* 0.091 · *nokia* 0.347	*sms* 0.104 · *auction* 0.235 · *you* −0.049 · *have* 0.019 · *won* 0.172 · *a* 0.082 · *nokia* 0.094
Integrated Gradients	*sms* 0.121 · *auction* 0.149 · *you* 0.037 · *have* 0.119 · *won* 0.266 · *a* 0.060 · *nokia* 0.244	*sms* 0.273 · *auction* 0.311 · *you* −0.105 · *have* 0.022 · *won* 0.201 · *a* 0.105 · *nokia* 0.171
Word Omission	*sms* 0.0005 · *auction* 0.001 · *you* 0.0001 · *have* 0.0007 · *won* 0.014 · *a* 0.0002 · *nokia* 0.008	*sms* 0.648 · *auction* 0.724 · *you* −0.012 · *have* −0.009 · *won* 0.556 · *a* 0.064 · *nokia* 0.175
Random	*sms* · *auction* · *you* · *have* · *won* · *a* · *nokia*	*sms* · *auction* · *you* · *have* · *won* · *a* · *nokia*

Table .3.: Evaluation of explanations for simple text which is classified as positive message from SMS Spam data set.

Methods	LSTM	Distill BERT
Prediction Probs.	[0.999999, 0.000001]	[0.9988, 0.0011]
SHAP	*i* $3.271e^{-5}$ · *can* $2.298e^{-5}$ · *take* $2.111e^{-5}$ · *you* $-7.228e^{-6}$ · *at* $3.574e^{-5}$ · *like* $3.887e^{-6}$ · *noon* $2.666e^{-5}$	*i* 0.000967 · *can* 0.00028 · *take* 0.00048 · *you* 0.00072 · *at* $9.202e^{-5}$ · *like* 0.00035 · *noon* 0.00062
LIME	*i* $-2.502e^{-5}$ · *can* $-1.683e^{-5}$ · *take* $-9.152e{-6}$ · *you* $9.175e^{-7}$ · *at* $-2.250e^{-5}$ · *like* $-2.858e^{-6}$ · *noon* $-1.415e^{-5}$	*i* 0.000234 · *can* $3.808e^{-5}$ · *take* $2.564e^{-5}$ · *you* $-4.105e^{-5}$ · *at* $6.749e^{-5}$ · *like* 0.000107 · *noon* $2.879e^{-5}$
Integrated Gradients	*i* $4.778e^{-5}$ · *can* $2.627e^{-5}$ · *take* $1.178e^{-5}$ · *you* $-6.001e^{-6}$ · *at* $3.656e^{-5}$ · *like* $7.619e^{-8}$ · *noon* $1.940e^{-5}$	*i* 0.000600 · *can* 0.000417 · *take* 0.000333 · *you* 0.000235 · *at* 0.000293 · *like* 0.000462 · *noon* 0.000275
Word Omission	*i* $5.837e^{-6}$ · *can* $1.814e^{-6}$ · *take* $6.637e^{-7}$ · *you* $-1.986e^{-7}$ · *at* $2.835e^{-6}$ · *like* $2.860e^{-8}$ · *noon* $1.109e^{-6}$	*i* -0.000115 · *can* -0.000118 · *take* -0.000146 · *you* -0.000252 · *at* 0.000103 · *like* $5.590e^{-5}$ · *noon* $4.744e^{-5}$
Random	*i* · *can* · *take* · *you* · *at* · *like* · *noon*	*i* · *can* · *take* · *you* · *at* · *like* · *noon*

Table .4.: Evaluation of explanations for simple text which is classified as negative message from SMS Spam data set.

B.3. Fake News

Methods	LSTM	Distill BERT
Prediction Probs.	[0.0000001, 0.9999999]	[0.0008, 0.9992]
SHAP	marine 0.0 · le 4.4703e−08 · pen 7.4505e−09 · tells 6.7055e−08 · off 2.9057e−07 · the 5.2154e−08 · two −2.9802e−08 · leaders −2.5331e−07 · who −7.0780e−08 · have 1.3038e−07 · accepted −4.097e−08 · refugees 7.0780e−08 · at −4.0978e−08 · the 3.3527e−08 · expense 1.3783e−07 · of −4.0978e−08 · the 2.2351e−08 · sovereignty 3.7252e−08 · of −4.4703e−08 · their −2.2351e−08 · nation 1.2665e−07	marine −0.0024 · le 0.0075 · pen 0.0015 · tells 0.0256 · off 0.00819 · the 0.00172 · two −0.00142 · leaders −0.00638 · who −0.00169 · have 0.01524 · accepted 0.00613 · refugees 0.00079 · at −0.0033 · the −0.0006 · expense 0.00282 · of −0.00494 · the −0.0007 · sovereignty −0.0012 · of −0.00417 · their 0.00558 · nation 0.00554
LIME	marine 3.8447e−5 · le 4.9912e−5 · pen 3.3469e−5 · tells 5.2299e−5 · off 6.9631e−5 · the 2.7412e−5 · two 2.4585e−5 · leaders −6.4858e−5 · who −8.4309e−6 · have 4.6953e−5 · accepted 1.9883e−5 · refugees 4.8676e−5 · at −1.1615e−5 · the 3.5340e−5 · expense 4.6186e−5 · of 9.6361e−8 · the 4.0473e−5 · sovereignty 3.2799e−5 · of −1.0208e−5 · their 1.1290e−5 · nation 4.3235e−5	marine −0.00025 · le 0.000255 · pen −7.2249e−5 · tells 0.0003 · off 0.0001 · the −9.1368e−6 · two 2.2283e−6 · leaders −0.0001 · who 4.1742e−5 · have 0.0002 · accepted 3.5926e−5 · refugees −5.6819e−5 · at −7.1489e−5 · the 9.6612e−5 · expense 8.2926e−5 · of −6.4577e−5 · the −3.16635e−5 · sovereignty 5.5631e−5 · of −5.6095e−5 · their −4.8520e−5 · nation −3.2682e−5
Integrated Gradients	marine 8.7826e−9 · le 6.0850e−8 · pen 1.5352e−8 · tells 8.7999e−8 · off 2.2736e−7 · the 2.2167e−8 · two −1.1039e−8 · leaders −1.3004e−7 · who −4.25953e−8 · have 1.0515e−7 · accepted −2.0024e−8 · refugees 6.3229e−8 · at −4.2262e−8 · the 1.5877e−8 · expense 1.0051e−7 · of −2.7647e−8 · the 1.3952e−8 · sovereignty 2.3178e−8 · of −2.4657e−8 · their −4.9468e−9 · nation 7.0227e−8	marine 0.0009312 · le 0.0030296 · pen 0.0022483 · tells 0.0031320 · off 0.0049658 · the 0.0035755 · two 0.0019714 · leaders 0.001090 · who 0.0026320 · have 0.0010148 · accepted 0.00087420 · refugees 0.00086188 · at 0.0013511 · the 0.0010993 · expense 0.0019455 · of 0.0011644 · the 0.001055 · sovereignty 0.0010229 · of 0.0014257 · their 0.0015607 · nation 0.003586
Word Omission	marine 0.0 · le 0.0 · pen 0.0 · tells 0.0 · off 1.1920e−07 · the 0.0 · two 0.0 · leaders −1.1920e−07 · who 0.0 · have 0.0 · accepted 0.0 · refugees 0.0 · at 0.0 · the 0.0 · expense 0.0 · of 0.0 · the 0.0 · sovereignty 0.0 · of 0.0 · their 0.0 · nation 0.0	marine −4.154e−5 · le −8.118e−5 · pen −0.0001772 · tells 0.0005291 · off 0.0006490 · the 6.330e−5 · two −2.890e−5 · leaders −0.0001688 · who −5.483e−6 · have 0.00038111 · accepted 1.746e−5 · refugees −0.0001214 · at 4.345e−5 · the −5.561e−5 · expense −4.345e−5 · of −4.750e−5 · the −2.533e−5 · sovereignty −0.00010699 · of −4.726e−5 · their −2.557e−5 · nation −2.723e−5
Random	marine le pen tells off the two leaders who have accepted refugees at the expense of the sovereignty of their nation	marine le pen tells off the two leaders who have accepted refugees at the expense of the sovereignty of their nation

Table .5.: Evaluation of explanations for simple text which is classified as fake(positive) news from Fake News data set.

Methods	LSTM	Distill BERT
Prediction Probs.	[0.999999 0.0000009]	[0.9998 0.0002]
SHAP	*moscow* 0.0023 *reuters* 0.9976 *russian* $-5.0678e^{-5}$ *president* $-9.1238e^{-6}$ *vladimir* $6.4029e^{-5}$ *putin* $-5.4191e^{-5}$ *said* $6.4677e^{-5}$ *on* $4.5842e^{-5}$ *friday* $2.9421e^{-7}$ *he* $1.9359e^{-7}$ *wanted* $-1.9970e^{-6}$ *constructive* $2.0083e^{-6}$ *relations* $1.0186e^{-6}$ *with* $2.2351e^{-7}$ *the* $-1.8990e^{-7}$ *united* $2.5931e^{-7}$ *states* $-2.7045e^{-7}$ *under* $3.5055e^{-8}$ *president* $-1.5615e^{-7}$ *elect* $7.4276e^{-7}$ *donald* $1.4670e^{-6}$ *trump* $-9.0400e^{-7}$	*moscow* 0.108966 *reuters* 0.733830 *russian* 0.0310 *president* 0.0214 *vladimir* 0.0058 *putin* 0.0062 *said* 0.0271 *on* 0.0083 *friday* −0.0076 *he* −0.0098 *wanted* −0.0237 *constructive* −0.0078 *relations* 0.0067 *with* 0.0006 *the* −0.0019 *united* 0.0147 *states* 0.0134 *under* 0.0067 *president* 0.0199 *elect* −0.0018 *donald* 0.0006 *trump* −0.0055
LIME	*moscow* −0.0193 *reuters* −0.9700 *russian* 0.0266 *president* 0.0113 *vladimir* −0.0026 *putin* 0.0136 *said* −0.0196 *on* −0.0044 *friday* 0.0031 *he* 0.0018 *wanted* 0.0177 *constructive* −0.0069 *relations* −0.0069 *with* 0.0035 *the* 0.00528 *united* 0.0026 *states* 0.0084 *under* 0.0051 *president* 0.0043 *elect* −0.0036 *donald* −0.0080 *trump* 0.0072	*moscow* 0.0688 *reuters* 0.767804 *russian* 0.0109 *president* 0.0066 *vladimir* −0.0233 *putin* 0.0180 *said* −0.0338 *on* 0.0221 *friday* 0.0169 *he* −0.0041 *wanted* 0.0209 *constructive* 0.0565 *relations* −0.0191 *with* 0.0090 *the* −0.0162 *united* 0.0005 *states* −0.0203 *under* 0.0190 *president* −0.0059 *elect* 0.0078 *donald* 0.0100 *trump* 0.0269
Integrated Gradients	*moscow* 0.1401 *reuters* 0.7486 *russian* −0.0877 *president* −0.0071 *vladimir* 0.0309 *putin* −0.0190 *said* 0.0851 *on* 0.0438 *friday* 0.0110 *he* 0.0057 *wanted* −0.0308 *constructive* 0.0338 *relations* 0.0231 *with* 0.0057 *the* −0.0034 *united* 0.0043 *states* −0.0046 *under* −0.0002 *president* −0.0014 *elect* 0.0127 *donald* 0.0212 *trump* −0.0119	*moscow* 0.1771 *reuters* 0.5208 *russian* 0.060 *president* 0.0580 *vladimir* 0.0212 *putin* 0.0176 *said* 0.0279 *on* 0.0205 *friday* −0.0096 *he* −0.0368 *wanted* −0.0211 *constructive* −0.0047 *relations* 0.0126 *with* −0.0044 *the* −0.0041 *united* 0.0139 *states* −0.0008 *under* 0.0005 *president* 0.0246 *elect* −0.0164 *donald* −0.0036 *trump* −0.0071
Word Omission	*moscow* $8.2279e^{-5}$ *reuters* 0.9999 *russian* $-8.2437e^{-7}$ *president* $-1.7390e^{-7}$ *vladimir* $1.2714e^{-6}$ *putin* $-3.4663e^{-7}$ *said* $8.2561e^{-6}$ *on* $2.0772e^{-6}$ *friday* $3.1120e^{-7}$ *he* $1.7880e^{-7}$ *wanted* $-5.0606e^{-7}$ *constructive* $1.3972e^{-6}$ *relations* $8.0939e^{-7}$ *with* $1.6328e^{-7}$ *the* $-6.4900e^{-8}$ *united* $1.2166e^{-7}$ *states* $-9.7030e^{-8}$ *under* $-1.2410e^{-9}$ *president* $-1.7390e^{-7}$ *elect* $3.8480e^{-7}$ *donald* $7.2930e^{-7}$ *trump* $-2.5408e^{-7}$	*moscow* $5.197e^{-5}$ *reuters* 0.448 *russian* $4.613e^{-5}$ *president* $1.788e^{-6}$ *vladimir* $-7.152e^{-7}$ *putin* $8.344e^{-7}$ *said* $5.364e^{-6}$ *on* $9.536e^{-7}$ *friday* $1.668e^{-6}$ *he* $5.960e^{-7}$ *wanted* 0.0 *constructive* $1.549e^{-6}$ *relations* $2.861e^{-6}$ *with* $5.9605e^{-7}$ *the* $-2.384e^{-7}$ *united* $2.622e^{-6}$ *states* $2.264e^{-6}$ *under* $9.536e^{-7}$ *president* $2.026e^{-6}$ *elect* $-1.430e^{-6}$ *donald* $-2.384e^{-7}$ *trump* $-1.1929e^{-6}$
Random	*moscow* *reuters* *russian* *president* *vladimir* *putin* *said* *on* *friday* *he* *wanted* *constructive* *relations* *with* *the* *united* *states* *under* *president* *elect* *donald* *trump*	*moscow* *reuters* *russian* *president* *vladimir* *putin* *said* *on* *friday* *he* *wanted* *constructive* *relations* *with* *the* *united* *states* *under* *president* *elect* *donald* *trump*

Table .6.: Evaluation of explanations for simple text which is classified as truth(negative) from Fake News data set.