Explaining the Predictions of Text Toxicity Classifiers

Nadia Sjöstedt

Department of Computer and Systems Sciences

Degree project 30 HE credits
Computer and Systems Sciences
Degree project at the master level
Spring term 2021
Supervisor: Ioannis Pavlopoulos

Abstract

Explainability mechanisms have opened exciting new doors in the machine learning community. Their purpose is to interpret the choices made by models during the prediction process, and explain these interpretations in a human-understandable manner. Such explanations allow users to understand the inner workings of complex models, something which can either increase trust or reveal a flawed logic. Models that process human language are faced with a challenging assignment, and pairing them with an explainability mechanism may therefore be of relevance. Toxicity detection is a task within the field of natural language processing that is becoming increasingly important with the rise of social media and user generated-content. The task involves detecting and classifying different forms of offensive language so that if needed, it can be removed. Numerous studies have examined text toxicity classifiers, and they have primarily done so by establishing their predictive performance in various tasks. The predictive performance of a classifier will indicate the extent to which it has assigned the correct class labels, but it will not reveal if it did so by interpreting the text correctly. In this context, explainability mechanisms can be used to expose any potential issues. The research problem addressed in this thesis is the lack of scientific projects which incorporate explainability in the evaluation process of text toxicity classifiers. The aim is to examine if predictive performance is a solid indicator of a model's suitability in toxicity detection. To do so, the following questions are answered: *What is the relationship between the predictive performance of text toxicity classifiers and the quality of the explanations they produce?,* with the sub-questions *Do text toxicity classifiers with a higher predictive performance produce explanations of higher quality compared to those with lower predictive performance?, Which out of the examined text toxicity classifiers produces the highest quality explanations?,* and *What are the properties of explanations provided by text toxicity classifiers?*.

The chosen research strategy is an experiment, the data collection method is observation, and the quantitative data analysis is done through statistical tests. Five text toxicity classifiers were evaluated, namely Naive Bayes, Logistic Regression, Random Forests, Long Short Term Memory, and Bidirectional Encoder Representations from Transformers. Each of them was paired with the mechanism Local Interpretable Model-Agnostic Explanations, and the explanations they produced were evaluated by comparing them against a dataset for which ground truth toxic spans are given. The findings indicated that the classifiers, despite having considerable variations in their predictive performance, bore some overall similarities when it came to their abilities to distinguish toxic words from non-toxic ones. However, the examination of how they distribute token weights showed that models with higher predictive performance are more reliable when it comes to assessing the toxicity level of individual words. Out of all models, Bidirectional Encoder Representations from Transformers is concluded as the one that produces the highest quality explanations.

Keywords: Toxicity classification, Explainability, Natural Language Processing, LIME, Machine Learning

Important notice: The thesis contains explicit language, reader discretion is advised.

Synopsis

Background
Explainability can be utilized to increase the trust in a machine learning model, but also to reveal issues. A model may have achieved high predictive performance with a wrongful logic, which makes it important to also inspect its inner workings. One field where explainability can bring value is Natural Language Processing. Toxicity detection is a task within this area that aims to identify toxic language online, and classification models are used for this purpose. In this context, explainability would help determine if the model has recognized toxic words when making predictions.

Problem
Studies that have examined toxic text classifiers mainly do so by establishing their predictive performance. This may be problematic since it does not establish whether their logic is sound. The research problem addressed in this thesis is the lack of studies that incorporate explainability in the evaluation of text toxicity classifiers.

Research Question
This study aims to establish if predictive performance is a solid indicator of a model's suitability in toxicity detection. To do so, the following questions are answered: *What is the relationship between the predictive performance of text toxicity classifiers and the quality of the explanations they produce?,* with the sub-questions *Do text toxicity classifiers with a higher predictive performance produce explanations of higher quality compared to those with lower predictive performance?, Which out of the examined text toxicity classifiers produces the highest quality explanations?,* and *What are the properties of explanations provided by text toxicity classifiers?*

Method
The chosen research strategy is an experiment, the data collection method is observation, and quantitative data analysis was performed using Spearman's correlation coefficient. Five classifiers were used, namely Naive Bayes, Logistic Regression, Random Forests, LSTM, and BERT, and their predictive performance was established through AUPRC, Precision-, Recall- and F1-scores in a text classification task. The explanations were generated for each of these using the explainability mechanism LIME, and the evaluation consisted of comparing them against ground truth toxic spans.

Results
Naïve Bayes obtained the lowest predictive performance in the text classification task, while BERT achieved the highest. As for the explanation scores, the LSTM achieved a high recall but low precision when detecting words, while Naïve Bayes did the exact opposite. The ability of the classifiers to assign appropriate weights to individual, toxic words was also evaluated. The results show that BERT was the strongest model, while Naïve Bayes was the weakest. The Spearman correlation coefficient also confirmed that classifiers with higher predictive performance tend to produce higher quality explanations than those with lower predictive performance.

Discussion
A limitation of the study is that a restricted number of text toxicity classifiers were evaluated. This was due to time constraints, and future work could therefore include exploration of a wider range of models. The main contribution of the thesis is that a greater understanding of the inner workings of text toxicity classifiers has been acquired, and that the association between their predictive performance and the explanation quality has been investigated.

Acknowledgments

No woman is an island, and I would like to say a few words about the individuals who made this paper what it is today. First of all, I feel very lucky to have had Ioannis Pavlopoulos as my supervisor. His knowledge, enthusiasm, and encouragement helped me go that extra mile. Thank you, Ioannis for always being available and supportive. I am also glad that we could borrow the brain of Panagiotis Papapetrou whenever needed, it was much appreciated. Continuing on the Greek thematic, I would like to thank my partner Antonis for never getting annoyed with me when holding endless monologs about explainability and toxicity. And for all the support, of course. I was also lucky to have Sonja, the best possible opponent. Finally, I would like to thank myself. It is not very Swedish of me, but I think it is befitting since I never thought I could do something like this, and yet I did.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AUPRC: Area Under the Precision-Recall Curve

BERT: Bidirectional Encoder from Transformers

CC: Civil Comments Dataset

FN: False Negative

FP: False Positive

LIME: Local Interpretable Model-Agnostic Explanations

LR: Logistic Regression

LSTM: Long-Short Term Memory

ML: Machine Learning

MSE: Mean Squared Error

NB: Naive Bayes

NLP: Natural Language Processing

P: Precision

R: Recall

RF: Random Forests

RNN: Recurrent Neural Network

SD: Span Detection

SD-F1: Span Detection F1

SD-MSE: Span Detection Mean Squared Error

SD-P: Span Detection Precision

SD-Recall: Span Detection Recall

TC: Text Classification

TC-AUPRC: Text Classification AUPRC

TC-F1: Text Classification F1

TC-P: Text Classification Precision

TCR: Text Classification Recall

TF-IDF: Term Frequency-Inverse Document Frequency

TN: True Negative

TP: True Positive

TSD: Toxic Spans Dataset

1 Introduction

1.1. Research Background

Understanding the logic behind predictions made by machine learning (ML) algorithms is crucial. Revealing a sound rationale gives credibility to the results, which in turn allows for well-founded decisions. When the logic is flawed, on the other hand, it permits corrective actions to be taken. This is the key purpose of explainable ML. In this context, an explanation can be described as a group of features that have led to a certain decision for a given instance (Montavon et al., 2018). A decision could, for example, be the assignment of a particular class label. Another characteristic of explainability is that the presentation of the model's inner workings should be self-supporting and understandable in human terms (Guidotti et al., 2018).

Many forms of explainability mechanisms have been developed, and they are used in various ML subfields. Natural Language Processing (NLP) is one such area, and the complexity and nuances in human language make explainability a key aspect of any NLP system (Lui et al., 2019). An example of the value it brings can be found in toxic language detection. The field has emerged due to the enhanced possibility for any individual to freely express their opinions, benign or otherwise, with a wide reach through online platforms. The downside of this freedom is evidently that toxicity in the form of harassment, provocations, or general disrespect has been given a free-for all-arena. The phenomenon has subsequently placed increased pressure on social media sites to address the problem, one way being to implement automatized detection and removal of malicious comments (Waseem et al., 2017). The purpose of explainability in this context is to identify which particular words led the classifier to predict toxicity (Risch et al., 2020), which reveals if the decision was just or not.

Undoubtedly, this task comes with challenges. To evaluate if a comment is correctly classified, a uniform understanding of what a toxic comment is must be established. Terms and definitions vary in the field, and toxic language classification is difficult to distinguish from related tasks such as hate speech-, abusive language- and cyberbullying detection (Aken et al., 2018). This may be seen as a challenge in creating a unanimous understanding. The definition of toxicity used to perform text classification in this thesis is the one proposed by Borkan et al., (2019, p. 491), which states that it refers to *"anything that is rude, disrespectful or unreasonable that would make someone want to leave a conversation".* What makes a comment toxic may come down to one word, or a span of multiple words which together create a toxic expression. In this thesis, a toxic span is defined as the part of the text responsible for the toxicity of a post (Pavlopoulos et al., 2021).

The choice of explainability mechanism should be done with the particular ML task in mind. Toxic language detection is an NLP task, and a study by Jeyakumar et al. (2020) determined that Local Interpretable Model-Agnostic Explanations (LIME) is the prominent mechanism for text analysis. Therefore, it is the mechanism utilized in this study. It was introduced by Ribeiro et al., (2016), and the authors describe its purpose as helping the user gain trust or to detect issues in the model being used. In this thesis, the model for which the behavior is being explained by LIME is referred to as *the base model*. *Local* in the acronym represents the fact that the given explanation should have a local fidelity, which means that it should correspond to how the base model would handle the original

instance being predicted. Local explanations are distinct from global ones where the goal is to explain the model prediction process as a whole, as opposed to for a specific prediction. *Interpretable* implies that the explanation must supply the user with a clear understanding of the relationship between the input and output variables (Garreau and Luxburg, 2020). NLP models often adopt techniques such as TF-IDF and word embeddings (Ribeiro et al., 2016), and in order to ensure interpretable explanations, this complexity must be hidden from the user. Therefore, the final explanation should only consist of the actual words which influenced the prediction. *Model-Agnostic* simply means that the algorithm should treat the model as a black box, and thereby be able to explain the predictions of any classifier or regressor (Garreau and Luxburg, 2020). This goes in contrast to model-specific explainability mechanisms which can only work with a certain type of model.

When classifying text, an explanation provided by LIME is essentially a list of the words in that text and their weights. The weights represent how much and in which way each word influenced the prediction probability. For instance, an explanation in toxicity detection may contain the word "stupid" assigned with a positive weight, which would indicate that the word had increased the prediction probability of the entire sample belonging to the toxic class. This is illustrated in Figure 1:

Figure 1 A toxic comment and the associated explanation provided by LIME

In Figure 1, all the positive weights, such as those of the words "stupid" and "terrorist" indicate that the words have contributed to the sample being assigned the positive class label, meaning the toxic class. Words with negative weights, such as that of "people" do the opposite, and their presence has increased the probability of the text contributing to the non-toxic class label.

LIME can, when used in parallel with evaluation metrics, be adopted as a complement to determine the best model for a certain context (Ribeiro et al., 2016). In a situation where one has to choose between two or several classifiers, it may not be sufficient to simply compare their predictive performance scores. Such metrics will give an indication of how well a classifier did in predicting the class labels of the samples. However, there is no transparency when it comes to knowing if it did so by interpreting the words correctly.

One perk of explainability in this context is the previously discussed possibility of detecting a wrongful logic which may still lead to high scores. An example of this is presented in the original LIME paper, where the authors use the mechanism to explain the predictions of a model classifying images as showing either a husky or a wolf. An accuracy of approximately 80% was reported, however, when inspecting the explanations an issue was revealed. The model was not using the parts of the images representing the animals, but instead, it was focusing on the background. Any image

containing snow was classified as a wolf, while all others were predicted as containing a husky. In toxicity detection, this could translate to using irrelevant parts of the post when making the prediction, or misinterpreting the meaning of individual words or spans.

The example brings to light the fact that classification scores are not always a sufficient means for assessing a classifier. Subsequently, this implies the necessity of observing the explanations of text classifiers and comparing their quality. In this context, the quality of an explanation refers to the extent to which the explainability mechanisms manage to accurately assess the text within a toxic post. In the case of LIME, this would involve the mechanisms recognizing toxic words as such by assigning them with weights that emphasize their toxicity.

1.2. Research Problem

Multiple studies have undertaken the task of evaluating text toxicity classifiers (D'Sa et al., 2020; Kajla et al., 2020; Leite et al., 2020; Saif et al., 2018;) and they have primarily done so by using the predictive performance of the classifier as a benchmark for evaluation. This is a recognized approach in the ML field, however, it does not allow to determine if the classifier reached the scores by interpreting the textual content in the instances the *right way*. Its logic may be flawed, but it can still end up with good predictive performance scores, as previously shown in the example with the classification of huskies and wolves. In toxicity detection, this may translate to the model attributing high importance to words that are not toxic, while missing or assigning low importance to those which in fact are, such as "idiot" or "dumbass".

Not knowing how the classifier reached a certain decision may be dubious in any machine learning subfield, but possibly even more so in a semantically complex area such as one dealing with text toxicity. The subtleties and nuances of human languages make it difficult to objectively assess if the textual explanation provided by some mechanism is reasonable or not. There are many ways of being implicitly rude or insulting, and the usage of explicitly profane words is not a precondition for toxicity. There are also algorithmic challenges that are general for the NLP field which may be even more prominent in this subfield, such as certain slang or misspellings not being in the vocabulary, multi-word phrases, and wrongful use of words (van Aken et al., 2018). Moreover, research has shown that the context in which a toxic post is found can both enhance or decrease the perceived toxicity (Pavlopoulos et al., 2020).

These are all complicating factors indicating that the task of assessing text toxicity classifiers must be performed with particular attention and that using only the predictive performance as a benchmark may not be enough. However, the tendency in the scientific community does not reflect this, and the assumption seems to be that predictive performance is a sufficient measure. Therefore, the research problem is defined as the lack of studies that go beyond using predictive performance to assess text toxicity classifiers by incorporating explainability in the evaluation.

1.3. Research Question

By addressing the research problem, the intention is to contribute valuable information to the field of NLP and toxic language detection. With this ambition in mind, the research question of the thesis is defined as follows:

What is the relationship between the predictive performance of text toxicity classifiers and the quality of the explanations they produce?

With the sub-questions:

1. *Do text toxicity classifiers with a higher predictive performance produce explanations of higher quality compared to those with lower predictive performance?*
2. *Which out of the examined text toxicity classifiers produces the highest quality explanations?*
3. *What are the properties of explanations provided by text toxicity classifiers?*

1.4. Research Objectives

The intent of this study is to systematically use explanations provided by LIME to assess the performance of different text classifiers. This allows a transparent understanding of the predictions since it brings light to which words and spans within the comment led to the determination of the class label. It also helps provide an understanding of the relationship between the performance of the classifier and the extent to which the explanations are sensible. The text toxicity classifiers evaluated in this study are Naïve Bayes (NB), Logistic Regression (LR), Random Forests (RF), Long Short-Term Memory (LSTM), and Bidirectional Encoder from Transformers (BERT).

Examining the explanations provided by LIME for the individual classifiers allows establishing the extent to which they are in fact detecting toxicity. Explanations were generated using a dataset where spans of toxic text within online posts had been determined by human annotators. This allowed using the parts of the posts marked as toxic by the annotators as ground truth values, which in turn enables assessing the sensibility of the explanations provided by the classifiers.

2 Extended Background

This chapter aims to provide the context and information needed to grasp the research presented in this thesis. Firstly, the key NLP tasks conducted in this project are described. Secondly, the text toxicity classifiers used in the study are presented with regard to their structure and documented performance in previous studies. Finally, explainability in the NLP domain and additional related work is discussed.

2.1. NLP Tasks

Natural Language Processing is a field in which various computational techniques are utilized in order to perform automatic analysis of human language (Young et al., 2018). The NLP tasks essential for the comprehension of this thesis will be discussed in the following subsections.

2.1.1 Text Representation

Machines are not able to read human languages in their natural form, and vector representations are therefore required to enable operating on them. Essentially, this means that the words and their significance must be transformed into numbers. There are many approaches for achieving this, Term Frequency–Inverse Document Frequency (TF-IDF) being one of them. The technique involves converting text documents in a dataset to a matrix representation, and the idea is to use the frequencies of all words in a collection of documents to measure the importance of each (Ramos, 2003). This importance is measured through a TF-IDF score, which is established by multiplying the term frequency by the inverse document frequency. The TF is equal to the number of times word X occurs in a text, and the IDF is equal to the total number of documents divided by the number of documents that contain the word X.

Word embeddings is another approach used to represent human language in a machine-readable format. While TF-IDF tries to capture the importance of a word through its frequency, word embeddings will focus on its meaning and relationships to other words. A vector representation is created for each word by training a model on a large corpus, and similar vectors may indicate that the associated words have a related meaning (Lai et al., 2016). This could allow detecting that words such as “jerk” and “idiot” refer to something highly related. There are many different forms of word embeddings, as well as ways of creating them. One can choose to create one’s own by learning them from a corpus by passing it in a tokenized form to an algorithm such as Word2Vec (Mikolov et al., 2013). Another option is to use pre-trained ones, where the embeddings have already been learned in a separate task. The idea is then that the captured information will still be applicable for solving other, similar tasks.

2.1.2 Text Classification

Text classification (TC) is a central task in the NLP field. It involves assigning a class label to a textual document based on the words, and main activities include extracting features, reducing

dimensionality, selecting classifiers, and evaluating the results (Kowsari et al., 2019). In this context, each word carries an indication of the meaning of the text, and aggregations are created to make a judgment concerning the overall character of the text as a whole. TC is performed using text classification algorithms. In supervised learning tasks, the algorithm builds a classification model by training on a set of labeled training samples, which allows capturing information regarding how the words in the document relate to the class labels. TC is widely used in areas such as document organization, opinion mining, email classification, and spam detection (Aggarwal and Zhai, 2012), and of course, toxic language detection.

2.1.3 Toxic Language Classification

Toxic language classification is an NLP task that involves building models to recognize textual content which in some way would make the reader want to leave the conversation. The models can then be used in automatic methods for detecting and removing abusive posts. A key benefit of toxic language detection is that comments and posts of a toxic nature can be removed, which contributes to a more friendly online environment.

The number of people who actively participate in the online community by sharing their thoughts and ideas is on a steady rise, and with that, the amount of abusive content. Mohan et al., (2017) investigated the relationship between toxicity and health in the online forum Reddit, with the results showing that toxicity always leads to a decline in forum health. Research also shows that adolescents are particularly vulnerable to this type of content, and that the effects can create mental health problems, and in the worst case even lead to suicide (Wijesiriwardene et al., 2020). Such serious indications evidently underline the importance of continued exploration.

The general approach in toxicity detection is to create a classification model by training it on a dataset containing examples labeled based on their toxicity. These labels may be binary, indicating that the instance is either toxic or non-toxic. They may also contain multiple classes, and reflect the severity or the type of toxicity in the post. Table 1 shows a number of samples from a dataset with binary class labels, where 0 represents non-toxic and 1 represents toxic:

text	toxic
Not sure where you got your definition of a good guy. You need to get a new dictionary. He was on the run from the law and has a very jaded past.	0
How did he pressure Kaneshiro?	0
I will bare my breasts after a brief statement. WTF? Starved for attention?	1
You are quite possibly the most offensively ignorant person to regularly haunt the Comments Section. What a piece of work. You have no bottom, it just goes farther and farther down...	1

Table 1 Toxic samples and non-toxic text samples with binary class labels

Table 1 shows how the human annotators have assessed the comments as either being toxic or non-toxic. However, detecting toxicity is not a straightforward task, and one of the difficulties lies in the fact that text toxicity is not a clearly defined phenomenon that can be established through some universal metrics. Tolerance levels and perceptions will vary between individuals. Another complicating factor is that the type of online conversations in which this form of language is found often contains complicating factors such as slang, spelling mistakes, and improvised shortenings of words (Gunasekara and Nejadgholi, 2018). Thankfully, non-toxic language is far more common than toxic language. This simultaneously means that the classifiers are acting on highly imbalanced data,

and with the toxic minority class being the positive one it can be considered a form of rare event prediction. Fewer examples of any class make it harder to predict, which generally leads to a classification bias towards the majority class (Fernandez et al., 2018).

2.2 Text Classification Algorithms

Various algorithms can be used when performing text classification, and the choice of which to use will depend on the task. Nonetheless, a general requirement is that the model should be able to effectively handle a large number of features with varying frequencies (Aggarwal and Zhai, 2012), as is the case with textual data. In this study, five text classification algorithms were evaluated, namely Naive Bayes, Logistic Regression, Random Forests, LSTM, and BERT. LSTM and BERT belong to the field of deep learning and will therefore be referred to as the deep machine learning classifiers. NB, LR, and RF will be referred to as the non-deep machine learning classifiers. The following subsections will provide an overview of the selected classifiers, including insights into their achievements in terms of predictive performance in previous research studies. Incorporating the latter in this study is of interest since the aim of this study is to understand the relationship between predictive performance and explanation quality.

2.2.1 Non-Deep Machine Learning Classifiers

Naive Bayes is seen as well-suited for NLP problems and is simple yet powerful in terms of accuracy at a low computational cost (Thangaraj and Sivakami, 2018). It has been used for document categorization since the 1950s and is to this day subject to research and development for the text classification task (Qu et al., 2018). Some of its main advantages are its suitability for text data, that it is easy to implement and fast to run, while a downside is its sensitivity to data scarcity (Kowsari et al., 2019). NB is proven to struggle in classification tasks on imbalanced datasets, a phenomenon which comes from the underrepresentation of one of the classes during the training process (Liu et al., 2009). This results in the classifier not having enough data points to draw from when learning to recognize the minority class. The mentioned phenomenon could make the algorithm a peculiar choice in the context of toxic language detection, seeing as imbalance data is the standard in toxicity detection. The inclusion of NB is made since this project does not aim to distinguish which classifier obtains the best predictive performance. Instead, the goal is to determine which classifier provides the most sensible explanations, and one cannot assume that there is a direct correlation between the performance and the explanations.

Logistic Regression is another suitable classifier for NLP tasks. Pranckevičius and Marcinkevičius (2017) have shown that it can outperform both NB and RF in a short-text sentiment classification task on online reviews. The study also demonstrated that all of the models manifested an insignificant increase in predictive performance when increasing the training set size from 5,000 samples per class to 75,000. An experiment by Kajala et al., (2020) used the same classifiers in a multi-class toxicity task, and LR once again came out as number one when evaluated using hamming loss, log loss, and accuracy. This indicates that it performs well in text toxicity tasks, and it is also straightforward to implement and requires little or no tuning. However, there are drawbacks. LR, much like NB, treats each data point as independent. Therefore, the classifier mainly predicts outcomes based on each word

as an independent feature (Kowsari et al., 2019). Intuitively, one can understand that this poses problems when it comes to text analysis. In most forms of exchanges and conversations, a word is not perceived as a stand-alone entity, but will rather get its complete meaning from other words in the conversation. LR and NB do not always take this into account, and examining the explanations may be an interesting way to determine how and if this issue presents itself.

Random Forests is often used in text classification due to its strengths in handling high dimensional and noisy data, and it has proven to yield good results (Islam et al., 2019). With that said, it is of importance to distinguish the classification of longer text documents from that of short-text, as is often the case with online comments. RF often struggles with short-text due to the sparseness of words and the often informal and varied use of words (Bouaziz, 2014), and its inferior performance in such situations has been documented (Pranckevičius and Marcinkevičius, 2017). Other factors which are essential to keep in mind are that, despite its fast training, it is slow in making predictions and it has a tendency to overfit (Kowsari et al., 2019). Despite these challenges, it is still a robust and widely used algorithm. A comparative study made by Hartmann et al., (2019) evaluated the performance of RF, NB, artificial neural networks, and K-nearest neighbor in representing human intuition in a text classification task using data from social media. The authors state that RF is underrepresented in text classification research, but that their results actually show that it performs well. It obtained the best results (alongside NB) for all datasets, and it is especially strong in three-class sentiment tasks.

A general observation is that, based on previous studies, Logistic Regression seems to be the strongest of the non-deep models when it comes to toxicity detection. The chosen algorithms are evidently not all that exists. Examples of other common and popular algorithms for text classification tasks are Support Vector Machines, Decision Trees, K-nearest neighbor, and Boosting and Bagging techniques (Kowsari et al., 2019), and the exclusion of them and others are merely due to the limited time-frame of this research project.

2.2.2 Deep Machine Learning Classifiers

Deep learning models generally achieve higher predictive performance than the ones discussed in the previous subsection, but a general issue is that this comes at the cost of lower interpretability (Kowsari et al., 2019). The models are exceedingly complex, and this black-box characteristic makes it difficult to grasp the logic behind the outcomes. This can, undoubtedly, have an effect on the user's level of trust.

With the aspect of the size of the dataset in mind, Ezen-Can (2020) conducted a comparative evaluation of an LSTM and a BERT model to determine their suitability for smaller datasets. The training set size used in the study consisted of 15,000 samples, and the results clearly showed that the LSTM is the better choice for a smaller corpus. This is especially interesting seeing as it is a lower computational cost than BERT. A study by Nowak et al., (2017) has also shown that the LSTM is especially suitable for short-text sentiment classification, such as online comments, and that bidirectional LSTM's perform better than the one-directional ones for all datasets used in the evaluation. Previous research also shows that both one-directional and bidirectional LSTM's are superior to Logistic regression when used for toxic language classification (Salif et al., 2018).

BERT, the second deep learning classifier used in this study, and there are a number of pre-trained models to accommodate various NLP problems. By the time of its publication, the authors could prove state-of-the-art results in a wide range of tasks (Devlin et al., 2018), and the success has continued.

The field of Toxic Language Detection is no exception. Research has shown that BERT is able to outperform a Bidirectional LSTM and a convolutional neural network when it comes to correctly classifying samples as toxic (d'Sa et al., 2020) as well as achieving top scores in a classification competition for classifying aggression in the short text (Gordeev and Lykova, 2020).

2.3 Explainable NLP

The NLP field has made substantial advances in recent years. Despite this, only a minority of the research studies conducted in the area have been dedicated to understanding the explainability aspect of NLP systems (Liu et al., 2018). This may be seen as problematic as the increased complexity of the models used in the field makes them more of a black box than ever. Adding explainability to a model can help build trust in the results, or alternatively, reveal issues. This makes it an important contribution to the entire ML field, for what good are the results if we cannot know whether or not we can trust them? Such uncertainty may lead to skepticism and a reluctance to put the models to use in a real decision process. On the other hand, a solid explanation can do just the opposite.

In a survey on the state of explainability in the NLP domain made by Danilevsky et al., (2020), the authors state that the standard way to categorize explainability is primarily based on two separate aspects. The first concerns whether the explanations refer to individual predictions of a model, or if it describes the model's predictive process as a whole. The former are known as local explanations, while the latter are known as global. The second aspect concerns if the explanation is generated naturally by the model, or if some post-processing is needed. This means that in addition to being global or local, and an explainability technique can also be either self-explanatory or post-hoc.

In this particular study, the area of interest is local explainability. On this level, explainability in text classification involves letting the user know how various words in a given text led to a particular prediction output. The words become the explanation, and they can often provide sufficient information to enable an interpretation of the model's behavior (Mathews, 2019). For instance, if an explanation for a post classified as non-toxic shows that the word "asshole" contributes to the non-toxic label, then this would indicate an issue in the model's ability to recognize the toxic language. Examples of self-explanatory local prediction methods are attention mechanisms and first-derivative saliency (Danilevsky et al., 2020). Both will assess the importance of individual words or combinations of them, and this information can then be used directly to establish which words carried the highest importance in establishing the class label. In that sense, these methods are explainable by design.

The other version of local explainability mechanisms is post-hoc, which entails adding some form of post-processing beyond the regular workings of the classifier. An example is a recently presented technique called Confident Itemset Explanation (CIE). Introduced by Moradi and Samwald (2021), it is the first approach to use confident item sets to represent the decision boundaries of black-box classifiers, both on the instance- and the class level. The itemsets are used to measure the strengths of the local relationships between the words, or a set of them, to the predicted label. The authors are able to show promising results, however, the mechanism is still in its infancy.

2.4 LIME

In a survey conducted by Jeyakumar et al. (2020), 455 participants were asked to specify which explainability mechanism they preferred for certain tasks, such as image, audio, and text. Six of the most popular explanation methods were put to the test, namely LIME, Grad-CAM++, Anchors, SHAP, saliency maps, and explanation by example. The results showed that 70% of all respondents thought that LIME was the best mechanism when it came to text, making it the preferred option in the area. These results indicate that it is a robust and appreciated mechanism, making it a suitable choice to explore further in this study.

2.4.1 Method

LIME is local and post-hoc. It was created by Ribeiro et al., (2016), and the explainability mechanism uses local, interpretable surrogate models to interpret predictions of a base classifier, such as text toxicity classifiers. The interpretable surrogate is used to approximate the inner workings of the more complex base model, which in turn allows explaining its behavior. When applied in TC, LIME attempts to interpret the base model by perturbing the input text sequence and then observing which effects this has on the prediction probabilities for a given instance. In this study, the instances are the toxic posts. The effects are established by calculating a similarity score between the original prediction probabilities and those produced when perturbing. The perturbed data points representing the words are removed one by one, and weights are then assigned based on how much the absence of each word affects the original output. The weight of a given word in a post should roughly correspond to the changes in the prediction probability which its removal evoked. For instance, if removing the word "moron" from a text resulted in the prediction probability of the toxic class decreasing from 75% to 65%, then its token weight should be set to roughly 0.10.

LIME's methodology differs from model-specific approaches since it does not require any understanding of the inner workings of the model. It allows establishing which words contribute to the final prediction output, and in which way. In binary toxicity classification, this involves establishing which words strengthen the probability of the text belonging to the toxic or the non-toxic class. Figure 2 shows a LIME explanation and how the different words in the post have influenced the prediction output:

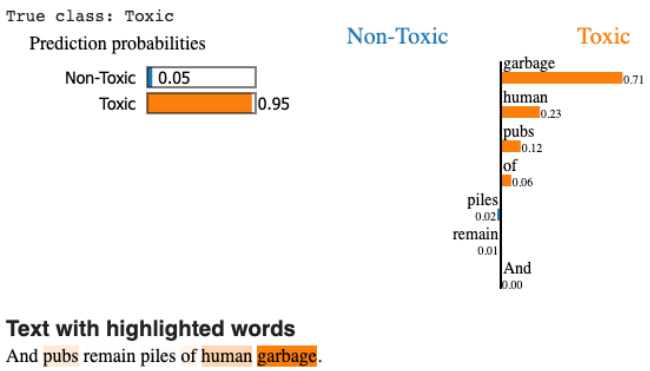

Figure 2 Explanation provided by LIME in a binary classification task

The true class of the instance explained in Figure 2 is toxic, and it is indicated that the words “human” and “garbage” (marked in orange) have contributed to the prediction of the toxic class. This seems reasonable, making it the type of explanation that could reinforce the user’s trust in the classification model.

2.4.2 Fidelity-Interpretability Trade-Off

According to Ribeiro et al., (2016), the explanation model created by LIME should be a reliable approximation of how the base model (the model which behavior we want to understand) predicts individual instances. For example, if the base model would deem a certain word in the post as highly toxic, then this assessment should be communicated by the explanation model as well. The authors describe this as the explanation model having local fidelity. Another key characteristic is the interpretability of the explanation. An explanation is interpretable if it provides the user with a qualitative understanding of the relationship between the sample as a whole and the prediction. Often, this involves limiting the number of features used in the explanation model. In the case of this study, these features are the words in the sentence. It may be cognitively challenging to interpret an explanation with a large number of words, and it can therefore be preferable to use only a selection of them when constructing the explanation. In practice, this could involve using only 6 out of 100 words in a text to create the explanation. The local explanation model with interpretability constraint is defined as follows:

$$explanation(x) = arg \min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

Ribeiro et al., (2016) define an explanation as a model $g \in G$, where G is a group of interpretable models. $\Omega(g)$ is a measure of the complexity of the explanation, which may be minimized by limiting the number of words used in the explanation. L is the loss, measuring how close the explanation is to the prediction of the original model. $f(x)$ is the probability that x belongs to a given class according to the base model. Πx is a proximity measure that is used to define the size of the neighborhood around x included in the explanation. Putting it all together, this gives $L(f,g,\pi_x)$, which is a measure of how unfaithful g is when estimating f in the locality of π_x. Optimally, there should be both local fidelity and interpretability. To assure this, $L(f,g,\pi_x)$, should be minimized and $\Omega(g)$ should be low enough for humans to understand it.

2.4 Other Related Work

This study aims to examine the relationship between the predictive performance of text toxicity classifiers and the quality of the explanations they produce when paired with LIME. The explanations are generated using the Toxic Spans dataset (TSD), which contains columns displaying toxic spans ground truth values. This column allows performing spans detection (SD), meaning that the ground truth column can be used to assess the extent to which LIME was able to detect spans of toxic words in the text. Comparing the generated explanations against the ground truth values thereby allowed assessing the quality of the explanations.

The TSD is described in further detail in Section 3.2.1. It was created by Pavlopoulos et al., (2021) for SemEval, an international workshop in semantic evaluation. The dataset served as the data source in an academic coding challenge where the goal was to detect toxic spans, and the contestants were

evaluated using an F1 score. The top contender obtained a score of 0.708 using a BERT ensemble approach utilizing both short and long spans (Zhu et al., 2021). Karimi et al., (2021) achieved an F1 score of 0.667 by combining a CharacterBERT with a Bag-of-Words model, while Chhablani et al., (2021) obtained an F1 score of 0.689 by using a RoBERTa. The latter also made the observation that their BERT-based models had a tendency to predict non-toxic offsets as toxic. In this workshop setting, the contestants were limited by the protocol in the sense that the ranking was based only on their F1 score. No restrictions existed in this thesis when it came to which and how many measures could be used to assess the abilities of the models to detect toxic spans. Therefore, this study can be seen a continuation of the work produced in the SemEval challenge.

3 Methodology

In this chapter, the research methodology with regards to its design and application is presented. Firstly, the chosen research design is described and motivated, followed by a deliberation of an alternative approach. Thereafter, the application of the research method is presented. The section covering the application also contains all explanations and definitions needed to interpret the findings of this study.

3.1 Research Design

Research Strategy

According to Denscombe (2010), the overall purpose of a research strategy is to provide guidance and support when performing a scientific project, and that the choice of such should be made with the research question in mind. It is also important to consider the feasibility of available methods; all may not be possible from a practical perspective, such as resources and competency. The research strategy chosen for this study is an experiment. It is well suited for investigating the relationship between and properties of some given factors, and is especially suited in contexts where exact measures are required (Denscombe, 2010).

In this study, the relationship to be examined is that between the predictive performance of classification models and the quality of the explanation they provide. The properties of the explanations themselves are also of interest, and establishing their nature is required to answer the final sub-question. One version of performing the experiment could have been to use a single classification algorithm. This could have involved manipulating its performance by tuning parameters, changing the amount of training data, etc., and then investigate which model gave the best explanation. However, the focus in this study is not on comparing the explanations given when using one and the same classification algorithm, but rather to compare those given by different ones.

The characteristics of an experiment are often associated with understanding a causal relationship between an independent and dependent variable (Johannesson and Perjons, 2014). In this study, this could have involved investigating if an increased predictive performance causes the quality of the explanations to increase. Or conversely, if good quality explanations cause higher predictive performance. However, causality was not examined explicitly. Instead, the relationship between the predictive performance and the explanation quality was observed by measuring the correlation between the two variables.

Data Collection Method

Another aspect to consider when designing a scientific study is which research methods to use. Johannesson and Perjons (2014) describe that while the strategy will provide high-level guidance of the project, the methods are hands-on in defining how to collect and analyze the data. Quantitative data are collected in this study, and the chosen data collection method is Observation. The name comes from the fact that data are created as the researcher observes the phenomenon of interest. Within this project, it refers to observing the classification performance of the classifiers and the quality of the explanations provided by LIME when paired with each of the classifiers. These are the data of interest when discussing the data collection method used in this study. The publicly available

datasets used to train and evaluate the algorithms are only a means to produce different scores and measures, and it is these values that we want to observe and draw conclusions from.

Observations can be conducted in different manners, and the approach chosen for this study was structured observation. It is suitable when the nature of the data is known and can be predefined through variables or measures that describe what the researcher is looking for (Given, 2008). In the case of this study, the desired data could be categorized into multiple scores and measures, which allowed creating unanimous observation schedules for logging the values. One observation schedule was created for documenting the predictive performance scores of the text toxicity classifiers, and another one was designed for documenting the scores and measures relating to the explanations created by each of the classifiers when paired with LIME.

Data Analysis Method

Quantitative data analysis with inferential statistics was adopted. Inferential statistics are commonly used to investigate whether there is a difference between two or more populations (Johannesson and Perjons, 2014). In this study, the application of statistical evaluation involved examining the relationship between the predictive performance of the classifiers and the explanations they produced using Spearman's correlations coefficient. It should be noted that this test will not disclose causality, but rather the nature and strength of the relationship. A closer description of the coefficient, as well as the application of the data analysis method, is given in Section 3.2.7.

3.1.1 Alternative Research Design

Alternative approaches for conducting the research project were contemplated, and the most carefully considered option was to conduct design science research. It is an approach that focuses on the development and scientific study of artifacts, with a focus on how these solve real, general, and relevant problems for users within a given context (Johannesson and Perjons, 2014). In this research project, the scientific study of artifacts entails systematically evaluating and comparing text toxicity classifiers with a focus on the explanations they provide. It should be kept in mind that design science is not a research strategy, nor is it a research method. It is a way of conducting research. This means that one cannot choose “experiment over design science” for instance, because there is no contradiction between the two. One can conduct a design science study, during which the experiment is used as the research strategy. With this in mind, the alternative research design would primarily have implied using the same building blocks in terms of research strategy, data collection method, and data analysis method as in the chosen research design, but with a design science approach.

According to Johannesson and Perjons (2014), a design science project must fulfill three overall qualifications. Firstly, a research strategy must be used for examining the problem at hand as well as for establishing the requirements of any stakeholders affected by it. Suitable methods must also be included for developing and/or evaluating the artefact/s of interest. When it comes to establishing the requirements, a focus group could have been used to collect data concerning the priorities of the stakeholders. For instance, questions such as what the most important aspects of the explanations are could have been discussed. Perhaps the stakeholders would find it more important that the explanations contain as many of the toxic words as possible, rather than the manner in which the weights are attributed to the individual, toxic words.

As for the evaluation, it would also have been possible to incorporate a more qualitative assessment of the explanations. The stakeholders could then have been asked to assess the explanations as images (such as the one previously displayed in Figure 2). This could have been done through a questionnaire containing explanation images from the different classifiers alongside questions regarding their interpretability and sensibility. However, after careful consideration, a more systematic take on eliciting requirements from stakeholders and adding a qualitative assessment was not seen as an optimal fit. Such a process is rigorous and time-consuming, and therefore not an evident choice when working under restrictions in terms of time and resources. It can also be argued that the strong focus on the elicitation of requirements from the perspective of stakeholders was not a main priority. In the current study, the superior artifact was to be established purely through metrics, and other functional or non-functional requirements were not of primary interest.

The second qualification of a design science project is that the results of the study must be compared and linked back to existing findings and knowledge in the research field. One of the purposes of this is to establish the originality and relevance of the findings. Comparing one's research finding to existing knowledge is an established approach, however, the originality or innovativeness of the artifacts compared in this study are not at the center. Instead, the focus is purely directed towards their ability to produce accurate explanations. The third condition of a design science research study is that the findings must be communicated to both the scientific community as well as the industry. This is a goal of most research projects, but may not be suitable as a strict requirement for one at the Master's level. With all these factors in mind, the design science approach was not adopted.

3.2 Application of Research Method

The methodology consisted of seven steps, starting with the selection of datasets and finishing with the statistical analysis of the results. Certain steps were iterative, such as going back and forth between the modeling- and evaluation steps when creating the classification models. Figure 3 shows the outline of the methodological procedure in its entirety:

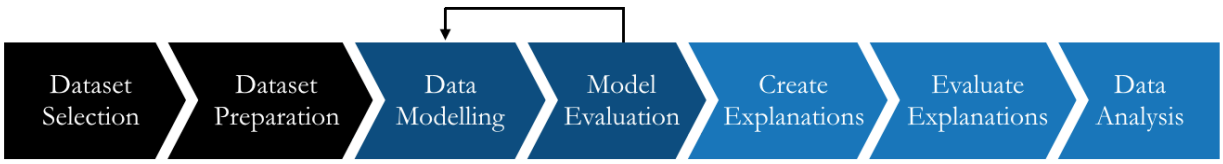

Figure 3 Outline of the steps in the research method

The subsections below present each of the steps in Figure 3 in closer detail. The preprocessing of the data and the experiments were conducted in Python using Google Colab, which is a web-based, integrated development environment.

3.2.1 Dataset Selection

Two publicly available datasets were used in this study; one for training and testing the classification models, and another for creating and evaluating the explanations. They are both presented and described below.

Civil Comments dataset

The Civil Comments dataset (CC) was used to train and test the text toxicity classifiers evaluated in this study, which allowed establishing their predictive performance in text classification. Created by Borkan et al., (2019), the dataset is publicly available through the TensorFlow library and it contains approximately 1.9 million samples.[1] Each sample consists of a post as well as features describing which type of toxicity the post contains, if any. The values of these features are floating-point numbers representing the fraction of annotators who found the sample to contain that particular kind of toxicity. These columns were all dropped since the type of toxicity is not of relevance in this study.

Each instance also contains the feature "toxicity", which represents a more general perception of toxic language. This column was used to create a binary class label; any instance which had a toxicity value greater than or equal to 0.5, meaning that a majority of the annotators had found it toxic, was assigned the value 1. All other instances were assigned with the class label 0, demonstrating that the post is non-toxic. The dataset is highly skewed, and approximately 8% of samples were assessed as toxic by at least half of the annotators.

Toxic Spans dataset

SemEval 2021 Task 5 provided the Toxic Spans dataset which contains 10,629 samples.[2] The posts in the dataset are derived from the Civil Comments dataset, and they are all toxic. The individuals annotating the TSD were not instructed to label the posts as a whole, instead, their task consisted of highlighting the actual parts of the post that they found to be contributing to its toxicity. This could be a single word, a span of words, or multiple spans of words. In this context, a toxic span is a sequence of words within a text responsible for the toxicity of that text. The dataset was chosen for this study due to the fact that it contains ground truth values, which allowed evaluating the LIME explanations. The dataset was not used to train and test the classifiers since it only contains toxic posts.

The TSD has eight columns, but only the text- and spans-columns were kept in their original form in this study. The text-column contains the toxic post, and the spans-column specifies the character offsets highlighted as toxic by the human annotators. A new column was also created using one of the original ones. The original column gave the probability of each character within the toxic comment being toxic, and represents the percentage of the annotators that marked that specific character as toxic. The new column represented the same thing but on the token level, meaning it specified the probability that each word was found toxic by the annotators. An excerpt of the three columns in the TSD utilized in this study are displayed in Table 2:

[1] https://www.tensorflow.org/datasets/catalog/civil_comments

[2] https://github.com/ipavlopoulos/toxic_spans/tree/master/data

	spans	text	token_probabilities
0	[29, 30, 31, 32, 33, 34]	”How about we stop protecting idiots and let nature add some bleach to the gene pool. We can always submit their names for the Darwin awards.”	{'How': 0.3333333333333333, 'about': 0.26666666666666666, 'we': 0.0, 'stop': 0.0, 'protecting': 0.0, 'idiots': 0.66666666666666, 'and': 0.0, 'let': 0.0, 'nature':...
1	[35, 36, 37, 38, 39, 40, 41, 42, 49, 50, 51, 52, 53, 54, 55, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72]	”Trump said, IN AS MANY WORDS, that Mexicans were rapists and drug dealers”	{'Trump': 0.0, 'said': 0.0, '': 0, 'IN': 0.0, 'AS': 0.0, 'MANY': 0.0, 'WORDS': 0.0, 'that': 0.0, 'Mexicans': 1.0, 'were': 0.0, 'rapists': 1.0, 'and': 0.0…
2	[ ]	”Trying the education route is best but you face a huge problem in that the average person today is dumber than a doughnut. Stupid, maybe you can do something but dumb, dumb is forever.”	{'Trying': 0.0, 'the': 0.0, 'education': 0.0, 'route': 0.0, 'is': 0.0, 'best': 0.0, 'but': 0.0, 'you': 0.0, 'face': 0.0, 'a': 0.0, 'huge': 0.0, 'problem': 0.0, 'in': …
3	[56, 57, 58, 59, 60, 61, 62, 63, 64, 65,66, 67, 68, 69, 70, 71, 72, 73, 74, 75,76, 77, 78, 79, 80, 81, 82, 83, 84, 85,86, 87, 88, 89, 90, 91, 92, 93, 94, 95,96, 97, 98, 99, 100…	“People are tired of seeing their countries overrun with illegal immigrants, crime, social parasites and welfare refugees and I applaud Hungary, Eastern Europe to protect their citizens and approve of LePen. Duterte is right killing druggies , but should be taken to court…	{'People': 0.0, 'are': 0.0, 'tired': 0.0, 'of': 0.0, 'seeing': 0.0, 'their': 0.0, 'countries': 0.0, 'overrun': 0.0, 'with': 0.0, 'illegal': 0.5, 'immigrants': 0.5, '': 0, 'crime': 0.5, 'social': 0...

Table 2 Excerpt TSD showing the ground-truth spans, toxic comments and token probabilities

In Table 2, the toxic words are marked in red for readability. The instance with index 0 is deemed to contain only one toxic word, namely “idiots”. The spans column indicates that the first letter of this word is found at index 29, and the last at index 34. The token probabilities column indicates that the toxicity of that word was corroborated by 66% of the annotators. The instance with index 1 contains multiple toxic words, and the token probabilities indicate that the word “rapists” was found toxic by 100% of the annotators. The text with index 2 is deemed as toxic as a whole, however, the annotators have not been able to point to any explicit toxicity in the post. Finally, the sample with index 3 contains multiple toxic spans.

3.2.2 Dataset Preparation

Preprocessing

Research has shown that thorough preprocessing of textual data in toxicity detection adds little or no value (Mohammad, 2018), and a moderate focus was therefore dedicated to this task. Unusable characters such as backlashes and question marks were removed without space, and stopwords were removed using the Natural Language Toolkit library’s stopword function, using the English language parameter.

Text Representations

TfIdfVectorizer from the Scikit-learn library was used to create the TF-IDF vectors in this study.[3] All parameters were set to default except for min_df and max_df. Both parameters work as thresholds to eliminate words that are either too uncommon and or too common based on their word frequency. Min_df establishes a threshold for minimum occurrence marking when not to include a word in the vocabulary, and was set to 8. Max_df is the maximum occurrence threshold and was set to the floating-point value of 0.9, meaning 90% of the documents. Both parameters were established through

[3] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

tuning, and the testing showed that changing the values did not have a considerable impact on the predictive performance scores on the TC level.

Pre-trained word embeddings were used as input to the LSTM classifier, and GloVe vectors were chosen. Created by Pennington et al., (2014) the approach aims to create a word-word co-occurrence matrix of all nonzero elements in the corpus. The type of pre-trained word vector used was GloVe's Common Crawl (42B tokens, 1.9M vocab, uncased, 300d vectors).[4]

Data Partitioning

A random subset of 400,000 instances from the CC was used for training and testing the classifiers in this study, and it was split into separate sets for training and testing. The training size was determined using learning curves, and the data was split using a 70:30 ratio. This meant that 70% of the samples were used for training and 30% for testing, resulting in the training set containing 280,000 samples and the test set 120,000 samples. Figure 4 shows the class distribution of the train set (left) and the test set (right), with 0 representing the non-toxic instances and 1 the toxic ones:

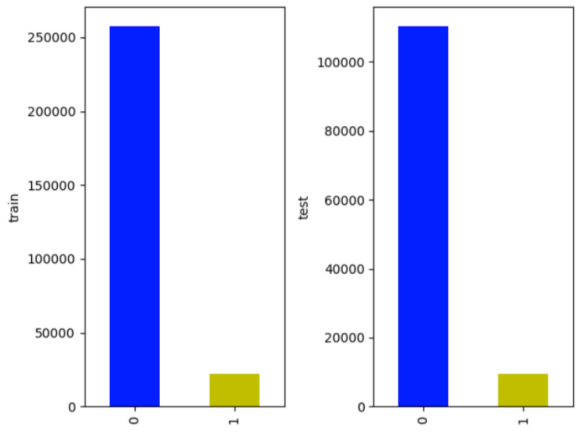

Figure 4 Class distribution in the CC train and test set

The CC is highly skewed as previously discussed, which resulted in there being 22,293 toxic instances and 257,707 non-toxic ones. As for the test set, there were 9,572 toxic instances and 110,428 non-toxic ones. Undersampling of the non-toxic majority class was tested to assess if this could positively affect the predictive performance scores. The approach did not yield an observable improvement and was therefore dismissed.

The TSD is already split into three separate categories; a training set consisting of 7,939 samples, a trial set consisting of 690 samples, and a test set containing 2,000 samples. The purpose of using the TSD in this study is purely to tune the parameters in LIME and to generate explanations, and therefore, only the trial and test set were used. The trial set was used to tune the parameters in LIME, while the test set was used to establish the final scores.

3.2.3 Data Modelling

Five different classification algorithms were used to conduct TC in this study, and their architecture and implementation are further described in this section.

[4] https://nlp.stanford.edu/projects/glove/

Naive Bayes

Naive Bayes uses the conditional probabilities of the Bayes Theorem. In toxic language detection, this entails calculating the probability that the class variable is toxic, given the words used in the particular comment being predicted. Each word is considered an independent feature, and those often found in comments labeled as toxic will contribute to a higher probability of toxicity, while words that do not will lower it. As described by Goodfellow et al., (2016), we can let X represent a vector of size n, $X_1 ... X_n$, which is to be classified into m classes, $C_1 ... C_m$, then we must find the probability of each class given X:

$$P(C_i|X) = \frac{P(X|C_i)\,P(C_i)}{P(X)}$$

Vector X, which in this study represents the TF-IDF vectors, is then assigned the class with the highest probability. The implementation from the Scikit-Learn library was used with default parameters, except for a random seed value of 42 for the random_state parameter.[5] The input was the TF-IDF vectors.

Logistic Regression

Logistic Regression is a supervised learning algorithm that splits the feature space linearly to create the best possible decision boundary to separate the samples. A logistic function is then used to predict the class label, and it has been defined as follows by Wright (1995):

$$logistic(\eta) = \frac{1}{1 + exp(-\eta)}$$

This function allows assuring that the output of a linear equation to be between 0 and 1. The probabilities of an example belonging to a certain class are derived based on its features, which are used to determine the class label based on some threshold value. If y is a feature, LR gives the probability of a binary output $y_i = \{0,1\}$ given input x_i (Kowsari et al., 2019). The Scikit-Learn version of the algorithm was used with default parameters, a random seed value of 42 was applied for reproducibility, and the TF-IDF vectors were passed as input.

Random Forests

Random forests is a supervised ensemble learning algorithm that combines multiple decision trees to create a single, predictive model. Each decision tree is trained on a random subset of the features, and each will create its own prediction. The best one is selected through a majority voting system, as shown in Figure 3:

[5] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

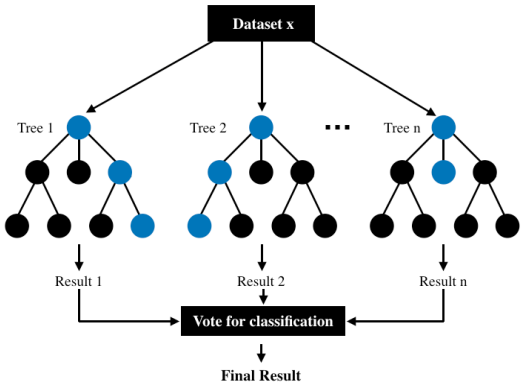

Figure 5 Random Forests prediction process

The idea behind combining multiple trees is that, even though some of them will produce errors, the majority will not. This allows moving away from wrongful predictions. The Random forest algorithm from Scikit-Learn was used with default parameters, the value of the random seed was 42 and the TF-IDF vectors served as input.[6]

LSTM

Created by Hochreiter and Schmidhuber (1997), Long short-term memory is a recurrent neural network which deals with the issues of short memory by using feedback gates in the form of neurons. The network is built in layers, each of which consists of recurrently connected memory blocks. The neurons rely on an activation function to make sure that all values in the network are kept in a range between 0 and 1, and these values are then used to determine which information to keep. Values close to 1 indicate that the information is important and should be kept, while those closer to 0 indicate that they may be forgotten. This architecture is suitable for operating on sequential data, such as text, and a cell state is used to carry relevant information throughout the different processing steps. In TC, this allows the network to learn which words within a text are important and which are not, and the information is used to make predictions. LSTMs can also be bidirectional (BiLSTM), which entails training a second LSTM on a reversed version of the input sequence. This gives the network access to even more information by providing both historical and future context.

The LSTM used in this study was built using the Keras library.[7] The first layer consisted of the embedding layer, and the GloVe word embeddings served as input. The next layer is the first of two LSTM layers. A stacked bidirectional architecture was used, meaning that these were two bidirectional layers. 100 neurons were applied in each. Two dense layers were used; one with 6 neurons and relu activation function, and another with 1 neuron and sigmoid activation function. Binary cross-entropy was selected as the loss function and the Adam-optimizer was used to find the weights. The batch size was set to 64, and training pursued during four epochs with early stopping. The input sequences were truncated to have a maximum length of 300, and pre-padding was used to assure that they were all of the same lengths. The choice of pre-padding was made based on testing, and on the documented benefit of choosing pre-padding over post-padding for LSTM's (Dwarampudi and Reddy, 2019).

BERT

Bidirectional Encoder Representations from Transformers is a bidirectional, transformer-based NLP model. The transformer is an architecture that uses attention-mechanisms in the task of transforming

[6] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[7] https://keras.io/api/layers/

one sequence into another (Vaswani et al., 2017), and it has made important contributions to the processing of human language. Building on the success of the transformer, BERT was introduced by Devlin et al., (2018) with the purpose of providing a pre-trained model that could be used in various tasks by just adding one additional output layer.

BERT uses a technique called Masked LM, which involves randomly concealing words or subwords in the post and then attempting to predict what they are. Using subwords is a way of handling tokens that are not in the vocabulary by breaking them down into smaller parts. For instance, one of the instances in the TSD contains the word "buttfuckers". If this token does not exist in BERT's vocabulary the model can break it down into the subwords "butt" and "fuckers", and interpret each of them separately. The model will try to predict the masked words or subwords by taking into account the words that come before and after at the same time. For instance, if you have the post "your dumb mother", this would involve using both "your" and "mother" when predicting the toxicity of "dumb". This approach allows BERT to use as much context as possible when making its predictions.

BERT is open source, and a large number of pre-trained models that only require fine-tuning are readily available. The one used in this study was the ELECTRA-small implementation from TensorFlow Hub, which is automatically mapped to the preprocessing model with the same name.[8] Other BERT models were tested, and out of those the chosen one obtained the highest predictive performance scores. One dropout layer and one dense layer were used, and sparse categorical crossentropy was selected as the loss function. The choice of loss function was made since it allows returning a 2D-array of prediction probabilities, which is required by the LIME algorithm.
Early stopping was used to determine the number of epochs, a patience of 2 was applied, which resulted in the model being trained during four epochs. As for the optimizer, Adam was chosen.

3.2.4 Model Evaluation

After having created the classification models, their respective performance in toxic TC was established using Precision (P), Recall (R), and F1 score, as well as the area under the precision-recall curve (AUPRC). Establishing these scores constitute the first part of the data collected in this study, and the P, R, F1, and AUPRC of each text toxicity classifier were logged in an observation schedule. The scores were denoted as Text Classification Precision (TC-P), Text Classification Recall (TC-R), Text Classification F1 (TC-F1), and Text Classification AUPRC (TC-AUPRC). This specification was introduced to avoid confusion since the measures were also used later to assess the quality of the explanations.

To understand these measures, the confusion matrix must be introduced. The matrix contains the following categories; True positives (TP) are instances that are correctly labeled as positive, false positives (FP) are negative instances incorrectly labeled as positives, true negatives (TN) are negative instances predicted as negative, and finally, false negatives (FN) refer to positive instances which are labeled as negative (Davis and Goadrich, 2006). Figure 6 shows a confusion matrix in a toxic detection task:

[8] https://tfhub.dev/google/collections/electra/1

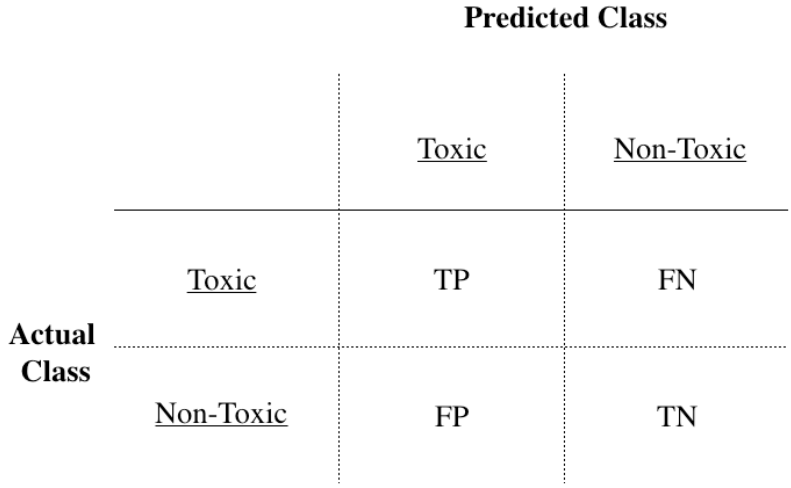

Figure 6 Confusion matrix for toxicity detection

Toxic is the positive class when dealing with toxic language detection, which means that the categories in Figure 4 correspond to the following descriptions:

- TP: Toxic is predicted, and the instance is in fact toxic
- TN: Non-toxic is predicted, and the instance is in fact non-toxic
- FP: Toxic is predicted, but the instance is non-toxic
- FN: Non-toxic is predicted, but the instance is toxic

A high number of FP's is problematic because it indicates that our classifier considers non-toxic comments to be toxic. In practice, this could lead to the automated removal of harmless posts. A high number of FN's on the other hand lead to an opposite issue, where toxic comments are labeled as non-toxic and therefore left unaddressed.

Precision and Recall

The categories in the confusion matrix can be used to calculate different performance metrics, and the choice of which should be made considering the problem at hand. As previously discussed, toxic language detection often involves a skewed class distribution with a disproportionately small number of toxic, positive samples. This indicates that we are interested in metrics that focus on the classifier's ability to predict positive samples. Precision and recall are examples of such. Based on a chosen threshold, precision establishes how many out of those predicted as positive are in fact positive, while recall determines how many out of those which are in fact positive have been correctly predicted as such. The definitions of precision and recall are presented as follows by Davis and Goadrich (2006):

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

F1 score

Ideally, we want to achieve both high recall and high precision. This indicates that our classifier is on point in detecting toxic instances while also recognizing positive instances as such. This is however not always the case. A classifier may have low recall and high precision, which means that it is overly picky in what it deems to be toxic. This results in it being precise in the predictions of positive instances, while the pickiness simultaneously leads it to miss many of them. High recall and low precision, on the other hand, indicate that the classifier is overly generous in handing out the positive class label. It predicts most toxic samples as toxic, but will, unfortunately, do the same for samples that are in fact non-toxic. This is a trade-off which for some ML-tasks makes it undesirable to observe the two metrics separately. In such cases, the F1 score can be a solution. It is the harmonic mean of precision and recall, and is defined as follows by Chinchor et al., (1993):

$$F1 = 2\,x\,\frac{Precision\ x\ Recall}{Precision + Recall}$$

The F1 score focuses on the classification of the positive class, making it especially suitable for measuring the classification performance in this study. It is measured in a range from 0 to 1, with 1 being the ideal. Obtaining high TC-F1 scores is a challenge in the text toxicity domain, and studies show that the scores are in the range of 15-60 percentage points that for predicting non-toxic posts (Zhang and Luo, 2018)

AUPRC

Precision-Recall curves is another suitable measurement for problems with a skewed class distribution, and it summarizes the trade-offs between precision and recall for various probability thresholds (Davis and Goadrich, 2006). The recall is plotted along the Y-axis and the precision on the X-axis and the visual representation of an ideal curve involves being in the top-right hand corner. The curve gives a visually intuitive indication of the performance of the classifier, but it may still be relevant to obtain a numeric metric to further represent what is seen. The area under the precision-recall curve can be used for this purpose, and as the name indicates, it is derived from the precision-recall curve and is simply calculated as the area beneath it. Essentially, an AUPRC of 1 means that the model has found all toxic samples and has not labeled any non-toxic samples as toxic. The worst possible AUPRC is 0.

3.2.5 Create Explanations

After having evaluated the models using the measures described in Section 3.2.4, each was paired with LIME to generate the explanations. It should be kept in mind that the models were trained and evaluated using the CC, while the explanations were created and evaluated using the TSD. Visualization techniques can be added to make the explanations more user-friendly as previously shown in Figure 2, but in its essence, an explanation provided by LIME for textual data is a weighted list of words. The weight of a word indicates its individual contribution to the prediction probabilities for a specific sample. In the case of toxicity detection, any word assigned with a positive weight is an explanation of what the classifier perceives as toxic. An example of such a list has been shown in Chapter 1, Figure 1. Since only the words with positive weights are relevant when evaluating a

classification model's ability to predict toxicity, all those with negative weights were filtered out in this experiment.

Implementing LIME: determining the number of words

The explain instance function in LIME's text module was used to generate the explanations.[9] The user decides how many words to include in the explanation using the num_features parameter, and it will take the *n* features with the highest token weights and include them in the explanation. As previously discussed in Chapter 2, the number of words will have an impact on the interpretability of the explanation model. In addition, it was also established that choice of *n* had an impact on the quality of the explanations when evaluated against the ground truth values in TSD. Choosing a high *n* resulted in false positives, seeing as tokens with evanescently small weights were included in the predicted spans. Setting $n = 1$, on the other hand, resulted in false negatives since LIME would be constrained from including more than one toxic word in the explanation.

The value of *n* was determined by calculating which integer value led to the explanation with the optimal fidelity for each instance in the dataset. In this context, optimal fidelity means that the explanation provided by LIME is the best possible approximation of how the base model, meaning the text toxicity classifier, would have predicted the given instance. This meant finding the *n* that would minimize $L(f, g, \pi_x)$, which has been previously discussed in Section 2.4.2. The approach for determining the value of *n* can be better understood through an example; assume that a base model predicts that the probability of a given instance being toxic is 60%. This would mean that the *n* leading to the optimal fidelity is the number of features whose summed weights have the smallest distance from 60. The optimal *n* was calculated individually in this manner for each instance in the TSD.

Besides num_features, only default parameters were used when implementing LIME. However, special considerations were taken when exploring the options for determining the settings of the kernel width and the bow parameters, both of which are discussed below. An overall inference from working with the LIME library is that it contains many tunable and possibly influential hyperparameters. A more profound immersement in this area was not conducted due to the limited scope of the research project, but doing so is recommended to explore in future work. This could involve performing a highly systematic tuning and evaluation of all the hyperparameters in the LIME text module, which could allow a more profound understanding of their individual impact on the explanations.

Implementing LIME: the kernel width parameter

The kernel width parameter defines the size of the neighborhood, meaning where the surrogate model is trained to approximate the base model for the instance being predicted (Molnar and Kopper, 2020). The size of the neighborhood is defined as π_x, and has previously been discussed in Section 2.4.2. The size is determined by applying weights to the tokens, and these weights are determined based on their similarity/proximity to the instance being explained. Choosing a kernel width to define this neighborhood in a correct manner is an unsolved problem in LIME (Zhao et al., 2020), and tuning was therefore performed to examine the effects. The tests showed that modifying the kernel width had only a minor influence on the results in terms of positive impact, and the parameter was thereby left at its default value of 25. A potential reason that the kernel width did not influence the scores considerably could be that changing this parameter did not modify the token weights to the point that the words went from being non-toxic to toxic, and vice versa.

[9] https://lime-ml.readthedocs.io/en/latest/lime.html#module-lime.lime_text

Implementing LIME: the bow parameter

The bow-parameter was another LIME feature that was explored in closer detail. When set to true, which is the default, a bag of words model is adopted. In LIME, this means that the input text sequence is modified so that only a single occurrence of each word exists, leading to the position of the word being ignored. When set to false on the other hand, the word position is considered, which can help highlight the fact that a word can be important or not depending on where in the post it is found. For instance, the word "cow" can be non-toxic when preceded by "cute", but toxic when preceded by "stupid". The bow parameter can be set to false for classifiers that use word order. Setting this parameter to false seems desirable for more effectively detecting toxic spans, however, doing so resulted in lower quality explanations for both BERT and the LSTM. Therefore, bow was set to true for all classifiers. No clear conclusions as to why this parameter did not improve the quality of the explanations of the BERT and the LSTM models were drawn, and this phenomenon is an example of what could be explored in future research.

3.2.6 Evaluate Explanations

The quality of the explanations was assessed using two forms of ground truth values from the TSD. Firstly, the spans column was used as ground truth to perform a binary evaluation. This means that a binary verification was made to check whether the character offset of the tokens identified by LIME also existed in the TSD spans column. If this was the case, it would indicate that LIME had detected the same words as the human annotators. Secondly, the token probabilities column in the TSD was used to perform a numeric evaluation. This means comparing the ground truth token probabilities against the weights assigned by LIME on the word level. Doing so allowed establishing how sensible the LIME weights were, the goal evidently being that these values should be as similar as possible to those established by the annotators. The two categories of evaluation are discussed in further detail below.

Binary Comparison

The explanations of each classification model were evaluated by comparing them against the spans column in the TSD. These spans have been established by annotators, and can thereby be seen as the ideal explanation concerning which parts of the text are toxic. In order to conduct a comparison between the explanations created by LIME and the offsets of the ground truth toxic spans, the explanations were converted into character offsets. These are shown in the LIME explanation column in Table 3. The metrics used in the evaluation were precision, recall, and F1 score on the character level. For clarity, these were denoted as Span Detection Precision (SD-P), Span Detection Recall (SD-R), and Span Detection F1 (SD-F1), and the evaluation can be understood by inspecting Table 3:

	spans	text	lime_explanation	SD-P	SD-R	SD-F1
0	[29, 30, 31, 32, 33, 34]	How about we stop protecting idiots and let nature add some bleach to the gene pool. We can always submit their names for the Darwin awards.	[29, 30, 31, 32, 33, 34]	1	1	1
1	[35, 36, 37, 38, 39, 40, 41, 42, 49, 50, 51, 52, 53, 54, 55, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72]	Trump said, IN AS MANY WORDS, that Mexicans were rapists and drug dealers	[35, 36, 37, 38, 39, 40, 41, 42, 49, 50, 51, 52, 53, 54, 55]	1	0.555	0.714

Table 3 LIME explanations with correct predictions (TP) in green, and missed words (FN) in blue

As shown in Table 3, the explanation created by LIME is an exact match to the target for the instance with index 0. This means that the explanation only contains true positives, resulting in perfect SD-P, SD-R, and SD-F1 scores of 1. For the sample with index 1, however, LIME has missed several words starting from offset 56 and upwards. This means that the prediction contains false negatives, namely "drug" and "dealers", in the sense that non-toxic was predicted when the words were in fact toxic. LIME did, for some reason, not assign positive weights to some of the words in the comment which the human annotators deemed as toxic. This resulted in an SD-R of 0.555 and an SD-F1 score of 0.714. Using this approach, a character-based evaluation could be established for the explanation of all instances in the test set. For each of the metrics, these values were summarized into a single, macro-averaged score, meaning that the mean of all evaluation instances was established.

The evaluation using the spans column as ground truth was done in a binary manner; the predicted offsets are either a match to those in the spans column, or they are not. Essentially, this will tell us if the classifiers when paired with LIME manages to recognize the same toxic words as the human annotators did. This is a relevant observation, however, it does not provide an understanding of the correctness in assessing the individual importance of those toxic words. For instance, this could help establish if LIME manages to recognize that the toxic word X is far more severe than the toxic word Y.

Numeric comparison

For the purpose of providing information concerning the importance of individual words, the token weights generated by LIME were compared against the token probabilities in the TSD. Optimally, the token weights should be close to the token probabilities. In order to establish if this was the case, the MSE was utilized, which is defined as follows by Mood et al., (1974):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

N represents the number of observations, and for each instance, this will be equivalent to the number of words in its toxic post. f_i corresponds to the token weight returned by LIME for word *i* in the toxic post, while y_i represents the actual value for word *i*, meaning its token probability (representing the fraction of annotators who included the word in their toxic spans). In the cases where the token weights were negative, meaning that LIME interpreted the words as non-toxic, the values were set to zero to enable comparison against the token probabilities. The MSE was denoted SD-MSE to maintain consistency between all scores that evaluate the explanations. Table 4 demonstrates an example of how the SD-MSE can be established for the toxic post "Screw you":

	Token probability	Token weight LIME	Sq. Error
'Screw'	0.66	0.45	0.0441
'you'	0.33	0.11	0.0484
		SD-MSE	**0.046**

Table 4 SD-MSE for a toxic post

For each toxicity classifier, the SD-MSE was established for each instance in the TSD. In addition, their respective mean SD-MSE was calculated across all instances in the dataset.

Explanation quality

The binary and the numeric evaluation approaches presented above were used side by side to assure that the quality of the explanations could be established. As previously discussed in Chapter 1, the quality of the explanations refers to the extent to which LIME is able to distinguish toxic words from non-toxic ones by assigning them with weights that reflect their toxicity (or lack thereof). The concept of evaluating the explanations using the chosen measures can be better understood through an example. Figure 7 demonstrates what would constitute an optimal explanation when using the chosen evaluation measures:

Toxic post

"You are a big fat idiot "

Ground truth spans

"You are a big fat idiot "

[14, 15, 16, 18, 19, 20, 21, 22]

Ground truth token probabilities

Word	Probability
You	0.0
are	0.0
a	0.0
big	0.0
fat	0.5
idiot	0.55

Predicted spans

[14, 15, 16, 18, 19, 20, 21, 22]

Predicted LIME weights

Word	LIME weight
You	-0.3
are	-0.4
a	-0.2
big	-0.1
fat	0.5
idiot	0.55

Explanation Scores

SD-P: 1

SD-R: 1

SD-F1:1

SD-MSE: 0

Figure 7 Example of a toxic post and its optimal explanation

Figure 7 illustrates the fact that an optimal explanation is one that corresponds fully to the ground truth values established by the human annotators. It should be kept in mind that the negative LIME weights representing the non-toxic words were set to zero when calculating the SD-MSE. LIME has thereby attributed token weights that are a perfect match to the token probabilities assigned by the human annotators, resulting in optimal explanation scores across all measures. Such accuracy is evidently not always the case, and there are many possible scenarios that could lead to lower scores. For instance, the token weights of the toxic words could have been higher or lower, resulting in a greater MSE. Non-toxic words such as "you" could have been mistaken for a toxic one, resulting in a lower SD-P and a higher SD-MSE. A toxic word such as "idiot" could have been recognized as non-toxic, resulting in a lower SD-R and a higher SD-MSE.

3.2.7 Data Analysis

The overriding focus of this study was to examine the nature of the relationship between the TC-scores and the SD-scores. The former represents the classification performance, and the latter the quality of the explanations. The procedure used to establish the predictive performance scores has been described in Section 3.2.4, and the approach used to represent the quality of the explanations has been

presented in Section 3.2.6. The purpose of this section is to clarify how the collected data was used to answer the research questions.

Main research question and sub-question 1

The main research question of this thesis is *What is the relationship between the predictive performance of text toxicity classifiers and the quality of the explanations they produce?* The following sub-question can be considered an elaboration of the main research question: *Do text toxicity classifiers with a higher predictive performance produce explanations of higher quality compared to those with lower predictive performance?* The liaison between the classification performance and the quality of the explanations was examined through Spearman's correlation coefficient, denoted as r_s. The coefficient measures the strength and direction of a monotonic relationship between two variables, and is defined as follows according to Artusi et al., (2002):

$$r_s = 1 - \frac{6\sum D^2}{n(n^2 - 1)}$$

D represents the difference between the two ranks of each observation, while *n* represents the number of observations. The observations consisted of the TC-F1 and the SD-MSE. The SD-MSE was chosen since it was considered the most essential score when it comes to assessing the explanations, seeing as it provides a higher level of detail compared to the binary evaluation measures. The higher level of detail comes from the fact that the measure will disclose how close to the ground truth the explanation came when assessing the level of toxicity of individual words. The binary scores, on the other hand, will only disclose to what extent the classifiers recognized the individual words as toxic or non-toxic.

It should be kept in mind that the optimal value for the SD-MSE is zero. Therefore, an r_s close to -1 would indicate that higher predictive performance on the TC level is associated with LIME assigning the token weights in a more sensible manner on the SD level. According to Corder and Foreman (2009), the relationship strengths associated with various values of the correlation coefficient can be interpreted as shown in Table 5:

Correlation Coefficient for a Direct Relationship	Correlation Coefficient for an Indirect Relationship	Relationship Strength
0.0	0.0	None/Trivial
0.1	-0.1	Weak/small
0.3	-0.3	Moderate/medium
0.5	-0.5	Strong/large
1.0	-1.0	Perfect

Table 5 Relationship strengths for the Spearman correlation coefficient

To ensure that the main research question could be answered in a precise manner, the null- and an alternative hypothesis were formulated as follows:

Null Hypothesis (H0): *There is no monotonic relationship between the predictive performance of text toxicity classifiers and the quality of the explanation they produce.*

Alternative Hypothesis (H1): *There is a monotonic relationship between the predictive performance of text toxicity classifiers and the quality of the explanation they produce.*

An alpha value of 0.05 was chosen to evaluate these hypotheses. Corder and Forman (2009) describe the alpha value as the probability of rejecting the null hypothesis by mistake when it is true. In order to assess if this is the case, a p-value is calculated. It allows evaluating the likelihood that the observed

relationship between the predictive performance and the explanation quality for the chosen sample have occurred by random chance. A p-value lower than alpha would indicate that the null hypothesis can be rejected.

Sub-question 2

The second sub-question was incorporated to ensure that it was established which classifier produces the best quality explanations, and is formulated as *Which one out of the examined text toxicity classifiers produces the highest quality explanations?* The SD-MSE was once again chosen as the most suitable measure to answer this question due to its elaborateness. Thereby, the classifier with the highest quality explanations is deemed to be whichever produces the lowest SD-MSE.

Sub-question 3

The purpose of including the final sub-question was to provide a more in-depth understanding of the individual explanations produced by each of the five classifiers, and was formulated as *What are the properties of explanations provided by text toxicity classifiers?* This was answered by individually examining all the explanation scores. The chosen metrics highlight strengths or issues that the classifier may have when making the predictions. For instance, a classifier with a low recall but high precision on the token level would indicate that it is picky and does not consider many words as toxic. However, when it does, it is usually on point.

3.2.8 Summary of Evaluation Measures

In summary, four different metrics were used to assess the classifiers in the TC task, and an additional four to evaluate the explanations. As a final data analysis, the r_s was calculated to establish the relationship between the TC-F1 and the SD-MSE. All measures starting with "TC" refer to an evaluation on the text level and have been established using the Civil Comments dataset. All those starting with "SD" refer to the evaluations on the spans level and have been established using the Toxic Spans dataset. In Table 6, all evaluation measures and their the optimal values are presented:

Measure	Description	Optimal Value
TC-P	The extent to which the instances predicted as toxic are in fact toxic.	1
TC-R	The extent to which the toxic instances have been predicted as toxic.	1
TC-F1	The harmonic mean of the TC-P and the TC-R.	1
SD-P	The extent to which the spans predicted as toxic are in fact toxic.	1
SD-R	The extent to which the toxic, ground truth spans have been predicted as toxic.	1
SD-F1	The harmonic mean of the SD-P and the SD-R.	1
SD-MSE	The MSE of the token weights assigned by LIME, and the ground truth token probabilities in the TSD.	0

Table 6 Overview of evaluation measures

In addition to the evaluation measures presented in Table 6, the r_s of the TC-F1 and the SD-MSE was established in order to answer the main research question. The description of this final analysis is described in Section 3.2.7.

3.3 Ethical Considerations

Ethics and a discussion thereof are a key part of any research project. The main focus in this area relates to potential participants in a research project; their interest must be protected, their participation voluntary and there must be transparency concerning the intended use of the data they provide (Denscombe, 2010). This study aims to compare and assess explanations provided by LIME when paired with a number of machine learning mechanisms. These classifiers are trained and evaluated on large datasets containing text collected from a commenting plugin for independent news sites. There is, in that sense, usage of data generated by humans, even though they have not directly participated in the study. These individuals have left comments on a public site, which made it possible for the creators of the CC to legitimately scrape and transfer them to a publicly available dataset containing approximately 1.9 million samples.

The reason why the usage of this data in the current research project does not call for any additional action is that it is anonymized. Nor the actual names or the usernames of the individuals posting the comments are included, which means that their identities are protected in the dataset. It should be kept in mind, however, that it theoretically could be possible to obtain the identities of these individuals using alternative measures. For instance, one could google one of the toxic comments and see if it lead to its original posting.

4 Results

This chapter contains the results of the thesis, and it is divided into three main parts. The first one presents the predictive performance scores of the classifiers, which consist of precision, recall, F1, and AUPRC on the TC level. Secondly, the evaluations of the explanations on the SD level are presented. As for the binary evaluation of the toxic spans identified by LIME, the scores comprise precision-, recall- and F1-scores, while the numeric evaluation involves establishing the MSE. Finally, the relationship between the predictive performance of the classifiers and the explanations they produce is established using Spearman's r_s.

4.1 Predictive Performance Measures

The text toxicity classifiers Naïve Bayes, Logistic Regression, Random Forests, LSTM, and BERT were trained and tested on the CC, and their predictive performance scores in text classification are presented in Table 7:

	TC-P	*TC-R*	*TC-F1*	*TC-AUPRC*
NB	**0.952**	0.041	0.080	0.469
LR	0.818	0.369	0.509	0.670
RF	0.751	0.411	0.531	0.632
LSTM	0.781	0.545	0.619	0.708
BERT	0.701	**0.755**	**0.726**	**0.806**

Table 7 TC Predictive performance scores of the text toxicity classifiers

The bolded values in the table indicate the best score for each evaluation measure. The predictive performance scores can be further interpreted by the clustered columns chart shown in Figure 8:

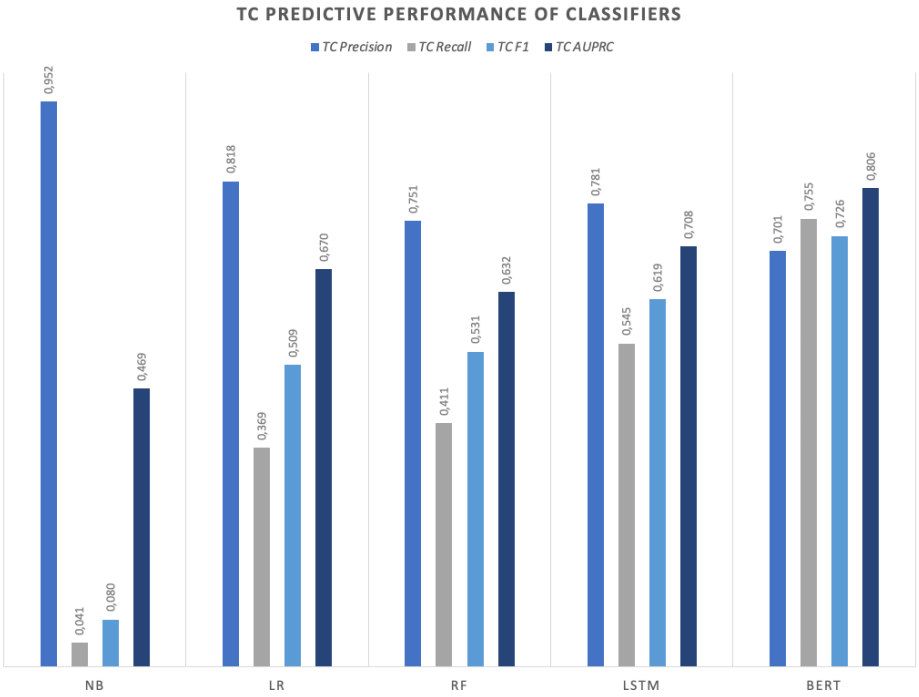

Figure 8 Visualization of predictive performance of classifiers

The TC-P of NB is higher than that of all other classifiers, while the TC-R is close to zero. This TC-P score indicates that the vast majority of the instances that NB deems to be toxic, are in fact toxic. There are few false positives, in other words. Meanwhile, its TC-R indicates that it has been overly picky in assigning the toxic class label, resulting in it wrongfully classifying toxic samples as non-toxic ones. This means that NB produces a large number of false negatives, which decreases the overall classification performance.

LR and RF follow a similar pattern as NB, meaning that the TC-P is higher than the TC-R. However, they both manifest considerably higher recall than NB, which subsequently results in better TC-F1 and TC-AUPRC. The LSTM also has higher TC-P than TC-R, but the gap between the two is less significant than for LR and RF. BERT is the classifier that obtains the best balance between TC-P and TC-R, and it also has the highest TC-F1 and TC-AUPRC.

4.2 Explanation Quality

After having trained and tested the classifiers, explanations were generated for each of them using LIME with the toxic spans test set. The evaluation was broken down into two categories, namely a binary and a numeric assessment, both of which are presented below.

4.2.1 Binary Evaluation

To enable the binary evaluation of the explanations, the words predicted as toxic by LIME were transformed into character offsets. These were then compared against the ground truth character offsets in the spans column of the toxic spans test set, and SD-P, SD-R, and SD-F1 scores were calculated for each instance in the test set. The macro averaged values of those scores are presented in Table 8:

	SD-P	*SD-R*	*SD-F1*
NB	0.550	0.683	0.536
LR	0.687	0.653	0.607
RF	0.669	0.601	0.578
LSTM	**0.774**	0.571	**0.612**
BERT	0.647	**0.700**	0.611

Table 8 Macro averaged SD-P, SD-R and SD-F1 of the classifiers

The binary explanations scores shown in Table 8 are visualized in the clustered column chart in Figure 9:

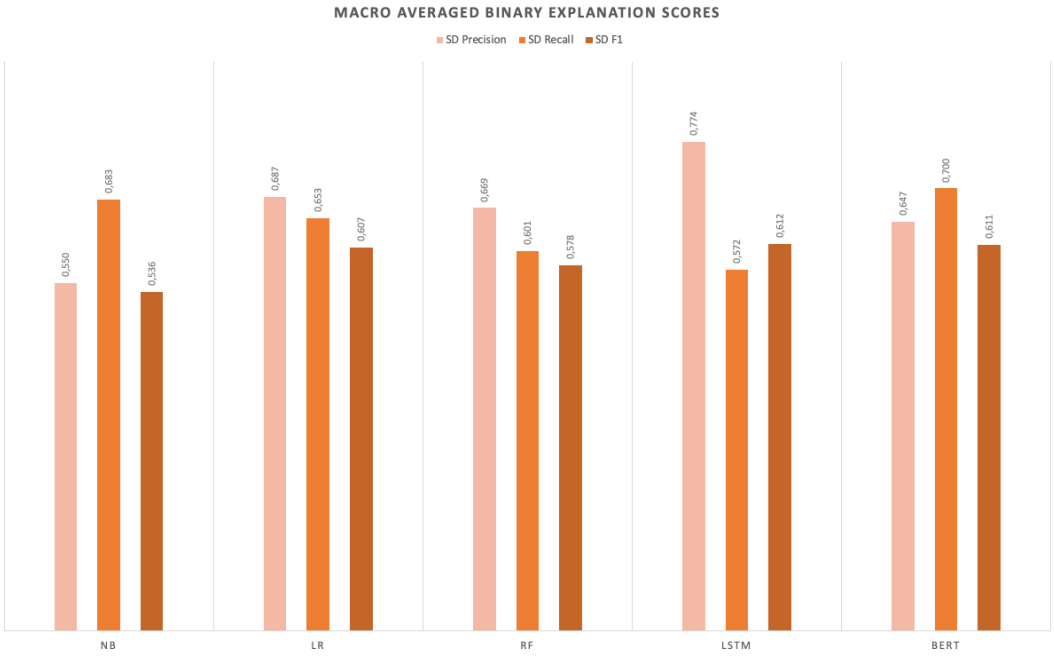

Figure 9 Visualization of the binary evaluation of the explanations on the SD level

In Table 8 and Figure 9, the SD-P of Naïve Bayes demonstrates that the model has a greater tendency than the others to misclassify non-toxic words in the posts as toxic. The SD-R on the other hand is the second highest out of all classifiers, meaning that it is relatively vigorous when it comes to assigning the correct class label to the target spans. The macro averaged SD-F1 of NB ends up as the lowest out of all the classifiers. Figure 8 also makes it evident that BERT follows a similar pattern as NB when it comes to span detection, with the difference being that the scores of BERT are higher.

The scores of Logistic Regression and Random Forest have a highly similar distribution, with the scores of the former being slightly higher. The allocation of their scores stands in opposition to those of NB and BERT in the sense that the LR and RF are more performant when it comes to SD-P than SD-R. This demonstrates that the token predictions that they made are correct to a higher extent than those of NB, but the lower levels of SD-R show that they also tend to miss some of the toxic words in the spans column, or entire spans. The LSTM also follows the distribution of having higher precision and lower recall. The SD-P is in fact the highest out of all classifiers, meaning that it is superior when it comes to recognizing the ground truth spans as toxic. The SD-R is lower, on the other hand, indicating that the LSTM also tends to miss toxic words when making its predictions. The LSTM and

BERT outperformed the other classifiers, with the LSTM obtaining the slightly higher SD-F1 out of the two.

4.2.2 Numeric Evaluation

The numeric evaluation of the explanations was conducted by comparing the token weights generated by LIME against the token probabilities in the toxic spans test set. This allowed establishing the SD-MSE for each instance in the test set, which was then summarized into a single score using the mean. Table 9 shows the mean SD-MSE of the token weights and the token probabilities of each classifier. It should be kept in mind that the desire is for the token weights to be as close to the token probabilities as possible, meaning that the lowest possible SD-MSE is preferred:

	MEAN MSE
NB	0.191
LR	0.170
RF	0.133
LSTM	0.140
BERT	**0.085**

Table 9 Mean SD-MSE of the classifiers

The mean SD-MSE values shown in Table 9 can be further interpreted through the column chart in Figure 10:

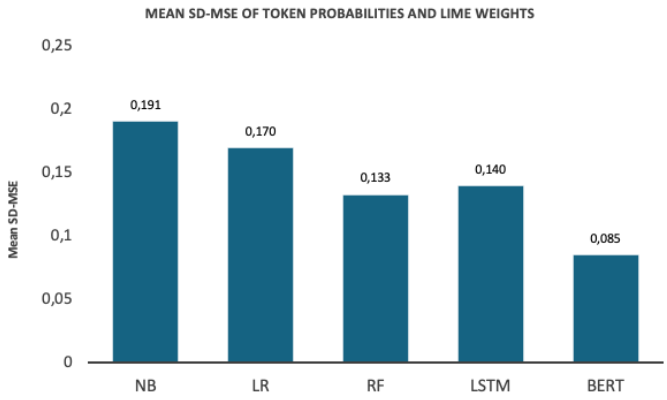

Figure 10 Visualization of SD-MSE of the classifiers

Table 9 and Figure 10 demonstrate that Naïve Bayes has the highest SD-MSE. This indicates that its manner of assigning weights, as interpreted by LIME, is the furthest from the ground truth values out of all the classifiers. Logistic Regression has a somewhat lower error than NB, while the errors of RF and LSTM are found at an even lower level. The classifier which assigns weights in a manner most similar to that of the annotators is BERT, which presents an SD-MSE of 0.085.

4.3 Predictive Performance and Explanation Quality

The relationship between the predictive performance of the classifiers and the quality of their explanations was established to answer the main research question. The Spearman correlation coefficient was established for the relationship between the TC-F1 and the SD-MSE, and these values have previously been presented in Tables 7 and 8 respectively. Table 10 shows them side by side, as well as the p-value and the final r_s:

	TC- F1	*MEAN SD-MSE*
NB	0.080	0.191
LR	0.509	0.170
RF	0.531	0.133
LSTM	0.619	0.140
BERT	0.726	0.085
	p-value	0.037
	r_s	**-0.899**

Table 10 Spearman rs of the TC-F1 and the SD-MSE

In Table 10, the TC-F1 represents the predictive performance of the classifiers, while the SD-MSE represents the explanation quality. The correlation is further demonstrated through the scatter plot shown in Figure 11:

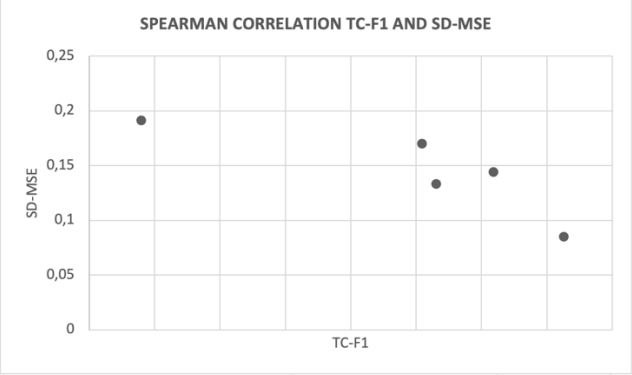

Figure 11 Spearman correlation between the predictive performance and the explanation quality

A p-value of 0.037 was established. This makes it lower than the chosen alpha of 0.05, meaning that the null hypothesis can be rejected and the alternative hypothesis accepted. The value of the coefficient is -0.899, and by consulting Table 5 it is established that this constitutes a strong/large indirect relationship. The fact that the coefficient is negative means that higher TC-F1 scores are associated with lower mean SD-MSE scores, and vice versa. In other words, classifiers that have higher predictive performance on the text level tend to assign token weights that are closer to the ground truth. Conversely, classifiers with lower predictive performance tend to assign token weights that are further from the ground truth.

5 Discussion

This chapter contains the discussion of the thesis, and firstly, the findings presented in the Chapter 4 are evaluated. Thereafter the research quality is debated with regards to strengths and potential limitations, and the societal and ethical consequences are also deliberated. Finally, suggestions for future work are given and the study as a whole is concluded.

5.1 Evaluation of the Findings

The first step of the data collection involved establishing the predictive performance scores of the classifiers on the TC level. Logistic Regression, performs relatively well in this task, something that has been documented in previous studies (Kajala, 2020; Pranckevičius and Marcinkevičius, 2017). In addition, the findings of Nowak et al., (2017) concerning the LSTM's suitability for short text sentiment classification was confirmed, as well as those of Gordeev and Lykova (2020) showing BERT's superiority over all other models when it comes to classifying aggression in short text.

It was also noted that all classifiers except BERT followed a trend of having a lower recall and higher precision on the TC level. This is aligned with the previously discussed statement made by Fernandez et al., (2018), namely that fewer examples of a certain class may lead to classification bias towards the majority class. In the CC dataset, the majority class is the non-toxic one. The low recall manifested by several of the classifiers is an indication that they tend to produce false negatives, meaning that non-toxic is predicted when the samples are in fact toxic. This is logical since the models have been exposed to much more non-toxic instances than toxic ones. As described by Liu et al., (2009), Naïve Bayes is especially challenged in this area, which was confirmed by a TC-R score close to zero.

However, the main focus of this thesis was not to establish one classifier's superiority over the others based on their predictive performance on the TC level. Instead, this was merely a means to the end of determining whether the predictive performance is indicative of the quality of the explanations of text toxicity classifiers. The second step therefore involved assessing the quality of the explanations, and the evaluation was broken down into two categories. The first approach was a binary assessment of the ability of the classifiers to detect toxic words and spans. As a point of reference, the highest-ranking F1 score in the SemEval contest was 0.708. It is considerably higher than the best SD-F1 in this thesis, being the LSTM's score of 0.612. This difference can to some extent be attributed to the context; the end goal for the SemEval contestants was to achieve the highest possible scores in terms of span detection. As for this thesis, however, high SD scores are not an end goal per se. Instead, the focus is on exploring the association between such scores and those obtained on the TC level.

As for the binary SD-F1 assessment, it became evident that the difference between the chosen classifiers was not substantial, and the scores were all within a fairly limited range; the lowest SD-F1 was 0.536, and the highest 0.611. This can be put in relation to the range of the TC-F1, where the lowest score was 0.080, and the highest 0.726. An important note, however, is that the TC was done using both toxic and non-toxic posts. The SD, on the other hand, was performed using only toxic posts due to the design of the TSD. This means that we cannot be sure what the scores would have been in case non-toxic posts had been incorporated into the dataset. The scores of the current span detection indicate that higher predictive performance in the TC task does not necessarily yield a proportionate

payoff in terms of improved span detection. At least not when conducting a binary assessment. As speculation, a possibility would be that including non-toxic instances in the TSD would lead to the strengths and weaknesses of the individual models coming through in a more distinctive manner. More precisely, that the stronger models would achieve even better scores on the SD level, while the weaker ones would do just the opposite. For instance, BERT has a strong classification performance in terms of both recall and precision, which indicates that it is more on point in distinguishing the two classes. Yet, in order to draw more firm conclusions concerning this aspect, the problem must be revisited using a dataset containing non-toxic posts.

The SD-P and the SD-R allowed giving a more nuanced understanding of the properties of the classifiers, since these scores manifested greater variations than the SD-F1. For instance, BERT and NB were superior when it came to the SD-R, while LR and the LSTM were the top players in the category of the SD-P. There can be many reasons for this, and it is beyond the scope of this thesis to establish what they are. Nonetheless, this makes an interesting starting point for future work.

Once the binary assessment was concluded, investigations were made concerning in what way individual words in the posts influenced the predictions of the classifiers. The values of the SD-MSE showed that, on average, the token weights assigned by NB were the furthest away from the ground truth probabilities, while BERT's were the closest. The other classifiers were scattered in between these two in terms of scores.

The values of the SD-MSE's become highly interesting when regarding them side by side with the SD-F1. As mentioned, the SD-F1 scores were fairly similar across the various classifiers, despite their TC-F1 varied greatly. When factoring in the SD-MSE, this suddenly makes more sense. It becomes evident that yes, a model with low TC scores such as Naïve Bayes can manage to distinguish a non-toxic word from a toxic one to a reasonable extent. However, where it falls short compared to the state-of-the-art is when assessing the importance of these words. For example, it may recognize that "idiot" is a bad word. It will therefore assign the token with a positive weight, however, this weight is very low in comparison to the one assigned by the human annotators. In that sense, the model is underestimating the toxicity of the toxic words. This may be an explanation as to why NB had a recall score close to zero on the TC level, which indicates that the model is missing a lot of toxic instances when assigning class labels. Underestimating the toxicity of individual words may naturally lead to the misclassification of the post as a whole.

The lowest SD-MSE was obtained by BERT, and thereby, the sub-research question asking which text toxicity classifier produces the best explanations has been answered. The SD-MSE of BERT was also considerably lower than that of RF, which came in as the second-lowest. Further validation of BERT's suitability in this task can be obtained by inspecting the SD-MSE alongside the SD-F1, which is also at the top of the class. In the same way that the high SD-MSE of Naïve Bayes was an explanation to its inferior predictive performance on the TC-level, the low error of BERT can potentially be seen as an answer as to why BERT is doing well when predicting toxic text. The way the model interprets the importance of individual, toxic words is more aligned with the perceptions of the human annotators, which subsequently makes it more pertinent in assigning the correct class label for an entire post.

The SD-MSE was chosen as the primary score to assess the quality explanations in this study. Therefore, the final data analysis involved establishing the Spearman r_s between the TC-F1 and the SD-MSE to allow answering the main research question and sub-question 1. The coefficient had a value of -0.899. By using the p-value it could be established that there is reason to believe that the

alternative hypothesis is true, meaning that *There is a monotonic relationship between the predictive performance of text toxicity classifiers and the quality of the explanation they produce.* The token weights assigned by models with a higher predictive performance on the TC level are more sensible than those assigned by models with a lower predictive performance. Essentially, this means that the quality of the explanations is higher when the predictive performance is higher, and vice versa. Finally, sub-question 3 involved understanding the properties of the explanations of the individual classifiers. This question does not have a short and clear-cut response. Instead, the answer is obtained by inspecting the figures and tables in Section 4.2.1- 4.2.2, and by following the elaboration of these in the argumentation presented in this discussion.

5.2. Research Quality and Limitations

Validity and reliability are important cornerstones of any research study, and the meaning of these concepts in quantitative studies has been discussed by Heale and Twycross (2015). The authors describe validity as the extent to which something has been accurately measured, while reliability is used to assess the accuracy of the instrument being used to make the measures. A desired attribute for the latter is that it produces consistent results when used repeatedly.

In the case of this study, assuring validity involves establishing that we are in fact measuring toxicity, and that this is done using well-established measures such as F1 score and MSE. The validity is strengthened by using the TSD as ground truth values. The dataset contains thousands of instances for which human annotators have distinguished toxic words from non-toxic ones, making it a solid benchmark. Validity is also reinforced through Observation, the chosen data collection method. It is regarded as having high validity, seeing as the process of making structured observations helps ensure structure and consistency in the measurements (Given, 2008). With that said, there is always a possibility of human error, and perhaps especially so when the research project is conducted by a single individual. Not even well-defined observation schedules can shield us from that, making it a possible restraint for the validity of this study.

A potential limitation of the study is the fact that there is a limited amount of data collected for the final analysis, a fact that became especially evident when inspecting the scatter plot in Figure 11. There are five observations in terms of the predictive performance of the classifiers, and another five representing the quality of the explanations. This may affect what is known as external validity, also known as generalizability. It refers to the extent to which the results can be generalized from the sample used in the study to the population as a whole (Ali and Yusof, 2011). The *population as a whole*, in this case, would refer to text toxicity classifiers in general. Incorporating more of them in the study could have made the findings more generalizable, and the choice of not doing so was purely pragmatic due to time restrictions. Expanding the study by adding additional models is recommended in future work.

As for reliability, the main area of interest is LIME's ability to create consistent explanations. An explanation by LIME is a local approximation by the surrogate model of the base model's behavior (Ribeiro et al., 2016). This way of approximating results in each explanation being unique. Concretely, this means that the token weights for the same instance being explained twice using the same base classifier will be slightly different. In this aspect, LIME can be seen as having a limitation by design when it comes to reliability if one chooses to take a highly critical mindset. With that said, the

differences between the individual explanations are evanescently small, and should therefore not be seen as considerable problems in terms of reliability.

A final, important aspect to keep in mind in quantitative research is reproducibility. In the machine learning field, it can be described as the ability to repeat a study with a high level of agreement in the results (Olorisade et al., 2017a). Reproducibility can never be obtained unless key details concerning the experimental setup are disclosed in the paper, and Olorisade et al., (2017b) have defined multiple aspects as important for text mining research studies; the datasets and how to retrieve them must be clarified, as well as how the data was preprocessed and partitioned for training and testing. The process of training the models should be disclosed, and relevant details concerning parameter setting should be presented including seed values for randomization. Finally, the software environment should be disclosed so that information concerning versions of the packages and modules used can be determined. All of the stated requirements have been accounted for in this thesis. In addition, links to all relevant implementations have been provided to enhance reproducibility. Despite having fulfilled these requirements, it is difficult to be completely assured that the results are fully reproducible. The study consists of many phases and components, and there is always a risk of one of them being interpreted incorrectly.

5.3 Ethical and Societal Consequences

Ethical and societal consequences may follow the findings of any research study, and it is therefore important to consider what such aspects could be for the project at hand. A recent report on such consequences in the domain of algorithms, data, and artificial intelligence was examined with the purpose of providing an in-depth understanding of the issue. In it, Whittlestone et al., (2019) describe these consequences as those that can impact different parts of society by either threatening or enhancing established values. It should be noted through this formulation that such consequences may be negative as well as positive. One category of consequences is defined as the one which involves automation and more efficient use of data resulting in individuals losing their livelihoods. Other technologies may be optimized for a certain group, while creating risks on a societal level. On the other hand, there is no doubt that various machine learning solutions have helped lift our society, such as those improving health care, home security, environmental protection, and so on.

The findings in this study can help improve the robustness of assessments of text toxicity classifiers. The positive effects of improving ways to correctly identify toxic language have obvious positive effects. Identifying and removing such language helps maintain a constructive online environment due to minimized exposure to malicious manifestations. In a day and age where we spend more time online than ever, this can be considered an important contribution. It is, however, important to keep the principles of free speech in mind in order to not overstep by conducting censorship. Toxicity detection operates with the goal of identifying and often also removing abusive language, and it is therefore important to be mindful of the boundaries in taking such activities too far.

5.4 Future Work

The work presented in this thesis opens multiple doors in terms of future work, and the most relevant ones are defined as follows:

Incorporate more text toxicity classifiers
A natural starting point for future work would be to expand the scope of this study by examining the explanations of more text toxicity classifiers. This could include well-established models such as Decision Trees and Support-Vector machines, but also additional BERT-based models. Doing so allows gathering more data, which subsequently provides more substance and generalizability to the results. Besides including more classifiers, it could also be of interest to perform a more systematic tuning of the ones already included.

Incorporate non-toxic posts in TSD
Another way of expanding the study would be to include non-toxic posts in the TSD. In order to simulate the natural underrepresentation of toxic language compared to non-toxic, such an approach could involve letting the toxic samples constitute the minority class.

Use the same methodology in another NLP task
In addition to exploring the problem even further in toxicity detection by adding more models, it would also be of interest to transfer the exploration to other areas of the NLP domain. The assumption that predictive performance on the TC level is a sufficient way to establish a superior model is surely not unique for the research area of toxic language detection. For instance, the exploration could be applied to sentiment classification on online reviews. The task would then involve establishing if the explanations provided by the models when paired with LIME did in fact manifest the sentiment to be expected for the established class label.

Improve classification performance of text toxicity classifiers
Another interesting exploration would be to use the findings as a diagnostics tool to help improve the individual models used in the study. For instance, the results show that the SD-R score of the LSTM model's explanations is relatively low. This sheds light on a shortcoming, namely that it has a tendency to misinterpret toxic words as non-toxic ones. Using this information insightfully can help guide efforts in how best to improve the model so that the overall classification performance can improve. Similarly, the SD-P of BERT is actually lower than that of LR, RF, and considerably lower than that of the LSTM. This evidently opens doors for further exploration and study, seeing as the model is considered state-of-the-art in the NLP community.

As a more general suggestion, future research could also consider gathering more information concerning how to best use LIME in the domain of text classification. LIME's implementation contains many hyperparameters, and understanding the impact of these is of interest considering the algorithm's previously discussed popularity in the NLP domain.

5.5 Conclusion

The aim of this study was to examine the relationship between the predictive performance of text toxicity classifiers, and the quality of the explanation they produce. To do so, an experiment was performed during which five classifiers were evaluated, namely Naïve Bayes, Logistic Regression, Random Forests, LSTM, and BERT. Firstly, their predictive performance was established, and

secondly, LIME explanations were created and assessed with regards to their quality. A dataset for which toxicity had been established on the spans- and token level allowed creating and evaluating the explanations. The execution of the research project allowed responding to the research questions, and the answers to these can be summarized as follows:

- There is a monotonic relationship between the predictive performance of text toxicity classifiers and the quality of the explanations they produce. Models with a higher predictive performance assign token weights in a manner that is more aligned with the ground truth values than models with a lower predictive performance.
- Text toxicity classifiers with a higher predictive performance produce higher quality explanations.
- BERT is the classifier that produces the highest quality explanations. This nomination was primarily based on the fact that it assigned token weights in the most accurate manner, however, it was also at the top alongside the LSTM when it came to predicting toxic words and spans.
- The properties of the explanations produced by the different classifiers vary. This thesis has shown that some of the models tend to misclassify non-toxic words as toxic ones, while others do just the opposite. Certain models are also more on point when it comes to attributing token weights that are more aligned with the ground truth values than others.

The findings indicate that low predictive performance on the text classification level does not necessarily translate to the model being poor in recognizing toxic words. For instance, Naïve Bayes demonstrated decent scores in span detection despite struggling considerably when it came to text classification. However, this discovery should be read in the light of there not being any non-toxic samples used in the creation and assessment of the explanations, as was the case during the text classification. A contributing factor to the variating predictive performance scores could be found when examining the way the classifiers assigned the individual token weights. The scores revealed that BERT was much closer than all other models when it came to assigning weights that were in alignment with the ground truth values, while Naïve Bayes had a relatively high error. Based on this, BERT was concluded as the model which produces the highest quality explanations.

The high-level contribution of this study involves having provided an increased understanding of the inner working of text toxicity classifiers. Explainability does just that, and applying LIME in a systematic and large-scale manner allowed collecting valuable data. By answering the research questions, the thesis has also helped shed light on how indicative the predictive performance of text toxicity classifiers is for the quality of the explanations. Possibilities for future work include incorporating more classifiers in the experiment and adding non-toxic samples to the TSD. Another area recommended for future exploration involves gaining a greater understanding of the LIME text module when used in toxicity detection. The parameter restricting the number of words used in the explanation was shown to have a considerable impact on the quality of the explanation, and this is an example of what could be explored further in future research.

References

Aggarwal, C. and Zhai, C., 2012. 'A survey of text classification algorithms'. In *Mining text data*, pp. 163-222. Springer, Boston.

Aken, B., Risch, J., Krestel, R. and Löser, A., 2018. 'Challenges for toxic comment classification: An in-depth error analysis'. *arXiv preprint arXiv:1809.07572*.

Ali, A.M. and Yusof, H., 2011. 'Quality in qualitative studies: The case of validity, reliability and generalizability'. In *Issues in Social and Environmental Accounting*, vol. *5*, pp.25-64.

Borkan, D., Dixon, L., Sorensen, J., Thain, N. and Vasserman, L., 2019. 'Nuanced metrics for measuring unintended bias with real data for text classification'. In *Companion Proceedings of the 2019 world wide web conference*, San Francisco, May 2019. Association for Computing Machinery, New York, pp. 491-500.

Bouaziz, A., Dartigues-Pallez, C., da Costa Pereira, C., Precioso, F. and Lloret, P., 2014. 'Short text classification using semantic random forest'. In *International Conference on Data Warehousing and Knowledge Discovery*, Munich, Germany, September 2-4, 2014, pp. 288-299.

Chhablani, G., Bhartia, Y., Sharma, A., Pandey, H. and Suthaharan, S., 2021. 'NLRG at SemEval-2021 Task 5: Toxic Spans Detection Leveraging BERT-based Token Classification and Span Prediction Techniques'. *arXiv preprint arXiv:2102.12254*.

Chinchor, N., Lewis, D.D. and Hirschman, L., 1993. 'Evaluating message understanding systems: an analysis of the third message understanding conference (MUC-3)', In *Computational Linguistics,* vol. 19, no. 3, pp. 409-449.

Corder, G.W. and Foreman, D.I., 2009. *Nonparametric statistics for non-statistician,* John Wiley & Sons, New York.

D'Sa, A.G., Illina, I. and Fohr, D., 2020. 'Bert and fasttext embeddings for automatic detection of toxic speech'. In *SIIE 2020 Information Systems and Economic Intelligence,* Tunis, Tunisia, February 6-8, 2020, pp. 1-5.

Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B. and Sen, P., 2020. 'A survey of the state of explainable AI for natural language processing'. *arXiv preprint arXiv:2010.00711*.

Davis, J. and Goadrich, M., 2006. 'The relationship between Precision-Recall and ROC curves'. *Proceedings of the 23rd international conference on Machine learning*, Pittsburg, USA, June 25-26, 2006. Association for Computing Machinery, New York, pp. 233-240.

Denscombe, M., 2010. *The good research guide: for small-scale social research projects*. 4th ed. McGraw-Hill Education, Glasgow.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. 'Bert: Pre-training of deep bidirectional transformers for language understanding'. *arXiv preprint arXiv:1810.04805*.

Dwarampudi, M. and Reddy, N.V., 2019. 'Effects of padding on LSTMs and CNNs'. *arXiv preprint arXiv:1903.07288*.

Ezen-Can, A., 2020. 'A Comparison of LSTM and BERT for Small Corpus'. *arXiv preprint arXiv:2009.05451*.

Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B. and Herrera, F., 2018. *Learning from imbalanced data sets*, Vol. 11, Springer, Berlin.

Given, L.M. ed., 2008. *The Sage encyclopedia of qualitative research methods*. Sage publications, London.

Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y., 2016. *Deep learning*. Vol. 1, No. 2, MIT Press, Cambridge.

Gordeev, D. and Lykova, O., 2020.'BERT of all trades, master of some'. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, Marseille, France, May 16, 2020. European Language Resources Association, Luxembourg, pp. 93-98.

Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, 2018. 'A survey of methods for explaining black-box models'. In *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–42.

Gunasekara, I. and Nejadgholi, I., 2018. 'A review of standard text classification practices for multi-label toxicity identification of online content'. In *Proceedings of the 2nd workshop on abusive language online (ALW2)*, Brussels, Belgium, October 31, 2018. Association for Computational Linguistics, Cambridge, pp. 21-25.

Hartmann, J., Huppertz, J., Schamp, C. and Heitmann, M., 2019. 'Comparing automated text classification methods'. In *International Journal of Research in Marketing*, vol. 36, no. 1, pp. 20-38.

Heale, R. and Twycross, A., 2015. 'Validity and reliability in quantitative studies', In *Evidence-based nursing*, vol. 18, no. 3, pp.66-67.

Hochreiter, S. and Schmidhuber, J., 1997. 'Long short-term memory'. In *Neural computation*, vol. 9, no. 8, pp.1735-1780.

Islam, M.Z., Liu, J., Li, J., Liu, L. and Kang, W., 2019. 'A semantics aware random forest for text classification'. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Turin, Italy, October 22-26, 2018. Association for Computing Machinery, New York, pp. 1061-1070.

Jeyakumar, J.V., Noor, J., Cheng, Y.H., Garcia, L. and Srivastava, M., 2020. 'How Can I Explain This to You? An Empirical Study of Deep Neural Network Explanation Methods'. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020),* Online Event, December 6-12, 2020.

Johannesson, P. and Perjons, E., 2014. *An introduction to design science*. Springer, Cham.

Kajla, H., Hooda, J. and Saini, G., 2020. 'Classification of Online Toxic Comments Using Machine Learning Algorithms'. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 13-15 May, 2020. Springer, Singapore, pp. 1119-1123.

Karimi, A., Rossi, L. and Prati, A., 2021. 'UniParma@ SemEval 2021 Task 5: Toxic Spans Detection Using CharacterBERT and Bag-of-Words Model'. *arXiv preprint arXiv:2103.09645*.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. and Brown, D., 2019. 'Text classification algorithms: A survey'. In *Information*, vol. 10, no. 4, pp. 1-68.

Lai, S., Liu, K., He, S. and Zhao, J., 2016. 'How to generate a good word embedding'. *arXiv:1507.05523.*

Leite, J.A., Silva, D.F., Bontcheva, K. and Scarton, C., 2020. 'Toxic language detection in social media for Brazilian Portuguese: New dataset and multilingual analysis'. *arXiv preprint arXiv:2010.04543*.

Liu, H., Yin, Q. and Wang, W.Y., 2018. 'Towards explainable NLP: A generative explanation framework for text classification'. *arXiv preprint arXiv:1811.00196*.

Liu, Y., Loh, H.T. and Sun, A., 2009. 'Imbalanced text classification: A term weighting approach'. In *Expert systems with Applications*, vol. *36*, no.1, pp. 690-701.

Mathews, S.M., 2019. 'Explainable artificial intelligence applications in NLP, biomedical, and malware classification: A literature review'. In *Intelligent Computing-Proceedings of the Computing Conference*, London, July 16-17, 2019. Springer, Cham, pp. 1269-1292.

Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. 'Efficient estimation of word representations in vector space'. *arXiv preprint arXiv:1301.3781*.

Molnar, C. and Kopper, P. 2020. *Limitations of Interpretable Machine Learning Methods*, viewed April 22, 2021, https://compstat-lmu.github.io/iml_methods_limitations/index.html

Mood, A.M., Graybill, F.A. and Boes, D.C., 1974. *Introduction to the theory of statistics, 3rd ed,* McGraw-Hill, New York.

Mohammad, F., 2018. 'Is preprocessing of text really worth your time for toxic comment classification?'- In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, Las Vegas, July 30-August 2, 2018. CSREA Press Inc., London, pp. 447-453.

Mohan, S., Guha, A., Harris, M., Popowich, F., Schuster, A. and Priebe, C., 2017. 'The impact of toxic language on the health of Reddit communities'. In *Canadian Conference on Artificial Intelligence*, Edmonton, May 16-19, 2017. Springer, Cham, pp. 51-56,

Montavon, W. Samek, and K.-R. Müller, 2018. 'Methods for interpreting and understanding deep neural networks'. In *Digital Signal Processing*, vol. 73, pp. 1–15.

Moradi, M. and Samwald, M., 2021.'Post-hoc explanation of black-box classifiers using confident itemset'. In *Expert Systems with Applications*, vol. 165, p. 113941.

Nowak, J., Taspinar, A. and Scherer, R., 2017. 'LSTM recurrent neural networks for short text and sentiment classification'. In *International Conference on Artificial Intelligence and Soft Computing*, Zakopane, Poland, June 3-7, 2017. Springer, Cham, pp. 553-562.

Olorisade, B.K., Brereton, P. and Andras, P., 2017a. 'Reproducibility in machine Learning-Based studies: An example of text mining'. In *ICML Reproducibility in ML Workshop at the 34th International Conference on Machine Learning,* Sydney, August 11, 2017.

Olorisade, B.K., Brereton, P. and Andras, P., 2017b. 'Reproducibility of studies on text mining for citation screening in systematic reviews: evaluation and checklist'. In *Journal of biomedical informatics*, *73*, pp.1-13.

Pavlopoulos, J., Laugier, L., Sorensen, J., and Androutsopoulos, I., 2021. 'Semeval-2021 task 5: Toxic Spans Detection' (to appear). In *Proceedings of the 15th International Workshop on Semantic Evaluation,* Online Event.

Pavlopoulos, J., Sorensen, J., Dixon, L., Thain, N. and Androutsopoulos, I., 2020. 'Toxicity Detection: Does Context Really Matter?'. *arXiv preprint arXiv:2006.00998*.

Pennington, J., Socher, R. and Manning, C.D., 2014. 'Glove: Global vectors for word representation'. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar, October 25-29, 2014. Association for Computational Linguistics, Cambridge, pp. 1532-1543.

Pranckevičius, T. and Marcinkevičius, V., 2017. 'Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification'. In *Baltic Journal of Modern Computing*, vol. 5, no. 2, pp. 221-232.

Qu, Z., Song, X., Zheng, S., Wang, X., Song, X. and Li, Z., 2018. 'Improved Bayes method based on TF-IDF feature and grade factor feature for Chinese information classification'. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Shanghai, China, January 15-18, 2018. IEEE Xplore Digital Library, pp. 677-680.

Ramos, J., 2003. 'Using tf-idf to determine word relevance in document queries'. In *Proceedings of the first instructional conference on machine learning*, vol. 242, no. 1, pp. 29-48,

Ribeiro, M.T., Singh, S. and Guestrin, C., 2016. 'Why should I trust you? Explaining the predictions of any classifier'. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, San Francisco, USA, August 13-17, 2016. Association for Computing Machinery, New York, pp. 1135-1144.

Risch, J., Ruff, R. and Krestel, R., 2020. 'Offensive language detection explained'. In *Proceedings of the Second Workshop on Trolling, Aggression, and Cyberbullying*, Marseille, France, May 11-16, 2020. European Language Resources Association (ELRA), Luxembourg, pp. 137-143.

Saif, M.A., Medvedev, A.N., Medvedev, M.A. and Atanasova, T., 2018. 'Classification of online toxic comments using the logistic regression and neural networks models'. In *AIP conference proceedings*, Maharashtra, India, July 5-6, 2018. AIP Publishing LLC.

Thangaraj, M. and Sivakami, M., 2018. 'Text classification techniques: A literature review'. In *Interdisciplinary Journal of Information, Knowledge & Management*, vol. 13, pp. 117-135.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I., 2017. 'Attention is all you need'. *arXiv preprint arXiv:1706.03762*.

Waseem, Z., Davidson, T., Warmsley, D. and Weber, I., 2017. 'Understanding abuse: A typology of abusive language detection subtasks'. *arXiv preprint arXiv:1705.09899*.

Wijesiriwardene, T., Inan, H., Kursuncu, U., Gaur, M., Shalin, V.L., Thirunarayan, K., Sheth, A. and Arpinar, I.B., 2020. 'ALONE: A Dataset for Toxic Behavior Among Adolescents on Twitter". In *International Conference on Social Informatics'*, Pisa, Italy, October 6-9. Springer, Cham, pp. 427-439.

Whittlestone, J., Nyrup, R., Alexandrova, A., Dihal, K. and Cave, S., 2019. *Ethical and societal implications of algorithms, data, and artificial intelligence: a roadmap for research,* Nuffield Foundation, London.

Wright, R. E., 1995. '*Logistic regression'.* In *L. G. Grimm & P. R. Yarnold (Eds.)*, *Reading and understanding multivariate statistics*. American Psychological Association, Washington, pp. 217–244.

Zhao, X., Huang, X., Robu, V. and Flynn, D., 2020. 'BayLIME: Bayesian Local Interpretable Model-Agnostic Explanations'. *arXiv preprint arXiv:2012.03058*.

Zhang, X., Luo, L. 2018. ‘Hate speech detection: A solved problem? The challenging case of long tail on twitter’. In *Semantic Web*, vol. 10, no. 5, pp. 925-945.

Zhu, Q., Lin, Z., Zhang, Y., Sun, Y., Li, X., Lin, Q., and Xu, R, 2021. ‘HITSZ-HLT at SemEval-2021 Task 5: Span-based ensemble model with toxic lexicon’, *SemEval*.

Young T, Hazarika D, Poria S, Cambria E, 2018. ‘Recent trends in deep learning based natural language processing’. In *IEE Computational intelligence magazine*, vol.13, no. 3, pp. 55-75.

Appendices

Appendix A – Reflection Document

• How does your study correspond to the goals of the thesis course? Why? Focus on the goals that were achieved especially well and those that were not well achieved.

I do believe that I have achieved the goals of the thesis project in a satisfying manner. It helped me a lot that I took this endeavor very seriously from day one, and I put a lot of effort into planning my work based on the course goals. One of the first things I did was to study the thesis instruction document closely. This allowed me to integrate it as a roadmap for as to how to conduct my project and achieve the goals. For me, it was important to write this paper independently and not in pairs. I wanted to feel that I was fully accountable for the final outcome and that I would have full ownership of all aspects. I also feel that I have grown a lot when it comes to searching, finding, and summarizing scientific literature. I have spent a great deal of time doing so throughout this project, and I believe that it has been one of the most meaningful parts of the course. Potentially a goal that could have been better achieved would be the one relating to analyzing and criticizing scientific literature. I do believe that I have done so mentally, especially when selecting which work to reference in my thesis. However, I have not necessarily expressed these thoughts in writing in my paper. At least not beyond the definition of my research problem, which of course can be seen as a criticism of previous scientific work.

• How did the planning of your study work? What could you have done better?

I am happy with how I planned my work. I wrote my Bachelor's thesis at DSV, meaning I was already familiar with the important steps and milestones in the thesis course. During the first week of the course, I made a week-by-week planning of the entire project. The planning was detailed, but not overly so since I was aware that things would change along with my learning curve. A also made sure that I had a good margin of time, in case some parts of the project took longer time than expected. The one thing I would do differently is to schedule starting the results chapter later than I did. I had a notion that I needed more time than I did to complete it, as well as the discussion, but I realized that this was not the case. I would have rather spent some more time tuning my models and improving my experimental setup rather than stressing over when to start writing the results.

• How does the thesis work relate to your education? Which courses and areas have been most relevant for your thesis work?

I would say that my thesis is highly related to my chosen specialization. The study contributes to the area of NLP, which is an important part of the Data Science domain. The courses that helped me the most in conducting this study are Programming for Data Science, Data Mining in Computer and System's sciences, Research Topics in Data Science as well as Scientific Communication and Research Methodology.

• How valuable is the thesis for your future work and/or studies?

I believe that it has been highly valuable. First of all, I have improved my overall skills in Python, and especially when it comes to handling and modeling large amounts of data. I also find it very useful that I now fully grasp the different phases in the machine learning/data science process. This has gone from being some abstract image on a lecture slide, to something that I have implemented myself over

and over again. Before I knew of these phases, now I understand them and they are in my long-term memory.

• *How satisfied are you with your thesis work and its results? Why?*

I am satisfied and proud of my work. I had very limited knowledge in the areas of machine learning and NLP before starting this course, but I still managed to produce something that I believe is of a nice quality. I used a divide and conquer technique, and made sure to do my best in every little part. I also believe that there is a level of innovation to this thesis. I have had a hard time coming across something similar, but I see it as obvious why this kind of research is needed. Explainability is growing, and it is only natural that different ways of using it keep evolving. I am also happy that the decision was made to use the measure the sensibility of the LIME weights by using the token probabilities as ground truth values. This was an idea that emerged later in the project, and I believe that it added much more nuance to the results.