NTNU
Norwegian University of
Science and Technology

Investigating the Effects of User Features in Hate Speech Detection on Twitter

Elise Fehn Unsvåg

Master of Science in Informatics
Submission date: June 2018
Supervisor: Björn Gambäck, IDI

Norwegian University of Science and Technology
Department of Computer Science

Abstract

Detecting hate speech has become an increasingly important task for online communities. Current methods for handling such unintended content are often heavily dependent on manual effort, and are therefore not scalable or efficient enough considering the large and growing corpus of user-generated data. Automatic hate speech detection is a challenging task, and a majority of the research in the field is targeting the task through the use of text. However, despite the emerging scientific studies to address the problem, there is still a need for further efforts to improve the quality and efficiency of detection methods. This motivates for research on how non-textual features can be utilized to enhance detection performance.

By using data from Twitter, the aim of this study was to investigate the potential effects user features have on hate speech classification. An overview of characteristic traits and promising user features was established through a study of related literature, and a quantitative analysis of Twitter data was conducted to better understand the characteristics of users, based on their tweets. The findings from the analysis show no correlations of hateful text and characteristics of the users who had posted them. Through experiments with a baseline hate speech classifier based on text on three different datasets, it was found that combining certain user features with textual features resulted in a slight improvement of classification performance. While the incorporation of user features resulted in varying impact on performance for the different datasets used, network-related features provided consistent improvement of the model performance. The inclusion of specific features was also found to be detrimental to the classifier performance, and some features only proved their usefulness when used in combination with others. These findings combine to suggest that there is a potential for incorporating user features to improve performance of hate speech detection models, but that more research should explore how they are most effectively utilized.

Sammendrag

Å indentifisere hatfulle ytninger har blitt en viktig oppgave for nettsamfunn. Nåværende metoder håndtering av hatytringer er ofte i stor grad avhengige av manuell arbeidskraft, og er derfor ikke skalerbare eller effektive nok med tanke på den økende mengden av brukergenerert tekstlig innhold på nett. Automatisk detektering av hatytringer er en krevende oppgave, og hovedandelen av forskingen gjort innen dette området har prøvd å løse problemet ved å se på teksten som blir skrevet. Til tross for den voksende andelen forskning som adresseser dette problemet er det fortsatt behov for ytterligere innsats for å øke kvaliteten og effektiviten av slike metoder. Dette motiverer for forskning på hvordan annen informasjon enn tekst kan bli brukt for å forbedre detektering av uønsket innhold på nett.

Målet med denne forskningen var å utforske de potensielle effektene brukerinformasjon kan ha i klassifisering av hatytringer, ved å bruke data fra Twitter. Fra en studie av relatert literatur ble det laget en oversikt over kjennetegn og brukeregenskaper som virker lovende å benytte i metoder for detektering. En kvantitativ analyse av data fra Twitter ble gjennomført for å bedre innsikt i egenskapene til brukere, basert på teksten i deres "tweets". Funnene fra analysen viser ingen relasjoner mellom hatefulle "tweets" og egenskapene til brukere som har skrevet dem. Eksperimenter med en baseline-modell på tre forskellige datasett viste at å kombinere visse brukeregenskaper med tekst bidrar til en liten forbedring av ytelsen til modellen som kun brukte tekst. Selv om de fleste brukeregenkaper hadde varierende påvirkning på modellen for de ulike datasettene, viste det seg at egenskaper relatert til en brukers sosiale nettverk hadde konsistent forbedring av modellytelsen. Noen egenskaper viste seg også å være ødelenggende for ytelsen, og noen egenskaper hadde kun positiv påvirkning når de ble brukt i kombinasjon med andre. Sammen antyder disse funnene at det er et potensiale for å innlemme informasjon om brukeren til å forbedre ytelsen til modeller for å detektere hatytringer, men at det er et behov for å forske mer på hvordan denne informasjonen kan benyttes på en effektiv måte.

Preface

This Master's Thesis was written as a part of the master degree program in Informatics at the Department of Computer Science (IDI), at the Norwegian University of Science and Technology (NTNU). I would like to thank Björn Gambäck for supervision and providing helpful feedback throughout the year. I would also like to thank the providers of the datasets used in this thesis, who have made significant efforts to collect and annotate data, and by sharing the datasets allow other researchers to make important contributions to the field of research.

Elise Fehn Unsvåg

Trondheim, 1st June 2018

Contents

List of Figures

List of Tables

1 Introduction

The Internet has provided the opportunity for easily expressing opinions and communicating, resulting in a massive amount of user generated data available for an enormous online audience. These opportunities also apply to those with malicious intentions, who can effortlessly and anonymously express hateful statements to large groups or targeting specific individuals. Identifying hate speech is a pressing issue for sites that allow user-generated content. Though there is not one formal definition, hate speech is commonly defined as abusive language that targets specific group characteristics, such as ethnicity, religion, or gender. The large and increasing amount of user-generated data on social media makes detection and removal of online hate speech difficult, which motivates for research in the field of how advanced technology can assist in solving the issue. This thesis will focus on how such technology can assist in automatically detecting hateful posts, by exploring information beyond the actual textual content. Specifically, information related to Twitter users will be investigated to evaluate the possible impact and effect this information has on hate speech detection.

1.1 Background and Motivation

Many online communities dedicate resources specifically towards removing hate speech or content violating their terms and conditions. In addition, communities also rely on their own users to report instances of hate speech. These methods depend on manual effort, and are therefore not scalable or efficient enough due to the large and growing corpus of user-generated data. Although online communities

share a responsibility to preserve freedom of speech, there is also a responsibility of preventing illegal hate speech online. Detecting and removing hate speech is important for online communities for maintaining safe environments for its users and as a responsibility considering their impact on society. This includes responsibilities such as protecting the many adolescent users of online communities that are more likely to be biased and affected by offensive content, removal of extremist content shared by terrorist groups, and removal of other offensive and hateful comments targeted at individuals or minorities. Twitter is one of the world's largest social networking services, with approximately 336 million active users[1], and is the platform that will be of focus in this thesis. Twitter has recognized the problem of malicious and abusive content, and has set a goal of making Twitter safer by detecting malicious automation, spam and fake accounts. However, Twitter itself along with several other researchers has yet to find effective methods for detecting abusive and hateful content.

There is an increasing amount of research covering the automatic recognition of online hate speech using Natural Language Processing (NLP). While most of the early studies focused on lexicon-based approaches for detecting "bad" words, the research field is expanding beyond this. Kwok and Wang (2013) found that 83% of their data was annotated racist due to the presence of offensive words. However, these approaches tend to have low precision because they mistakenly classify all messages containing specific terms as hate speech, and this is particularly challenging on social media sites due to the relatively high prevalence of offensive words (Wang et al., 2014). After all, hate speech can be much more sophisticated than that. There are several challenges faced concerning the task of hate speech detection, which will be presented in Chapter 3. One of these challenges includes finding the information or features that best represent the underlying phenomenon of hate speech. Existing studies have mainly focused on content-based text classification using features such as the appearance or frequency of words, spelling mistakes or semantic meaning. While these methods perform relatively well, there is still need for improvements to increase the quality of detection. Researchers have expressed the need for exploration for contextual information as well, such as information

[1] https://investor.twitterinc.com

about the users.

Information about the users could be either known factors, such as age and gender, or factors derived from behavior. There exists research that investigates the impact of different features, and research about the personality and behavior of users expressing hate speech. However, there is little research that combines the two topics. This is highly motivating for research on the subject, and the focus of this thesis will therefore be on user features and their effect on classification methods. Hate speech detection is of interest to many actors, and the motivation for this project is to contribute to the field of study, which can assist in making detection faster and more accurate.

1.2 Goals and Research Questions

Goal *Investigate the effect of user features in hate speech detection*

The goal of this research project is to investigate information related to users in the Twitter community that can be helpful in identifying online hate speech, and then experimenting with the features in hate speech classification. Both a theoretical and practical approach will be adopted to achieve this goal, with a literature study and with experiments. This objective is further divided into the research questions below.

Research question 1 *What does the literature and data suggest as promising user information for hate speech detection?*

To obtain an overview of user information that seems promising to use in classification, a review of related studies about users posting hate speech online will be conducted. The findings from this review will be analyzed and provide an understanding of which user information can be used as features in the experiments with hate speech classification, which is the focus of the second research question. An analysis of Twitter users in three different datasets will also be conducted to

identify any possible distinguishing characteristics of users who have posted hateful content and users who have not, based on their tweets.

Research question 2 *What are the effects of incorporating user features in hate speech classification?*

First, a hate speech classifier based on text will be implemented to serve as a baseline for comparison and measuring of the effects of various user features. The available and retrieved user features will then be systematically incorporated to the text classifier. Training and testing the classifier with the inclusion of user features on several datasets allows a deeper insight into the problem and larger grounds for comparison of the findings.

1.3 Research Method

To answer the research questions and accomplish the overall research goal, several methodologies have been used. First, a study of the findings from existing academic literature about users and online hate speech was conducted, where the findings went through a qualitative analysis to provide answers to Research Question 1. In addition to the literature study, the work conducted in this thesis follows an experimental research strategy to investigate the effects of user features in methods for hate speech classification. Based on the findings from the literature review, a statistical analysis of the users in the datasets used provided insight into possible relevant features to use in the experiments. The experiments were designed to provide answers to Research Question 2, by incorporating user features to an implemented classifier and comparing the results to a classifier using only textual features. The results of the experiments were qualitatively analyzed by evaluating the impacting factors of the classifiers and investigating the distribution of correctly and incorrectly predicted tweets. Although this research follows a experimental strategy, the main contribution of this thesis will not be to prove or disprove a hypothesis. Through an exploratory design, the work in this thesis will gain experience and insight regarding user features in hate speech detection.

1.4 Contributions

The work in this thesis will mainly contribute to a deeper insight into a field that is still in the need for more research, and that can be used to build upon for other researchers. An increased amount of research in this area will hopefully contribute to improving the methods for removing hateful content that are currently adapted by online communities. More specifically, the research conducted in this thesis will contribute with the following:

C1 A literature review on the subject related to users of online communities and hate speech.

C2 A quantitative analysis of characteristics of Twitter users, with a comparison of users in different target classes based on their tweets.

C3 The implementation of a baseline hate speech classification system with textual and user-related features.

C4 Experiments with textual and user-related features in hate speech classification to investigate the effects of individual features and feature subsets.

1.5 Thesis Structure

Chapter 2 introduces the relevant theoretical concepts and methods that are used in this thesis, or in related work.

Chapter 3 provides an overview of the research conducted in the field of hate speech detection, feature investigation and studies related to the authors of hate speech.

Chapter 4 presents the datasets of tweets used to train and test the implemented hate speech classifier. This is followed by an analysis of the characteristics of the users in the datasets.

Chapter 5 includes the experimental setup, and describes the architecture of the classifier developed and the experiments conducted to measure the impact of user features.

Chapter 6 addresses the research questions with an evaluation and discussion of the experimental results.

Chapter 7 concludes the thesis by summarizing the research contributions along with suggestions for potential future work.

2 Background Theory

This chapter covers the theory within the fields of Machine Learning and Natural Language Processing (NLP) that is relevant for this thesis and in related work on automatic hate speech detection. The basic concepts of the most common approaches are introduced in this chapter, while the next chapter will present the state-of-the-art. Lastly, tools and libraries used in this thesis are described.

2.1 Machine Learning Methods

Machine learning is a field of computer science that enables programs to learn from experience. Supervised learning is a type of machine learning where algorithms learn functions from labeled data, to be used in mapping new and unseen data. Methods for detecting hate speech mainly follow supervised learning approaches, and the task is mainly considered a classification problem. This section introduces the terminology and concepts of the common machine learning classifiers used for hate speech detection.

2.1.1 Logistic Regression

Logistic regression (LR), or binary logistic regression, is a simple classification algorithm that uses statistics to make predictions where the outcome is binary. The goal of LR is to find the best fitting model that describes the relationship between the outcome and a set of independent variables. The LR model estimates the

confidence of an outcome based on the independent variables by using a non-linear function called the *logistic function*, shown in Equation 2.1. The logistic function, also known as the *sigmoid function*, can take any real valued input and return an output in the interval [0,1], which can then be interpreted as a probability. The outcome represents the model's confidence in the classification, where values close to 1 indicate the first class and values closer to 0 indicate the other class. LR is also equivalent to *maximum entropy*, which according to Ratnaparkhi (1997) in NLP is a model that combines different contextual evidence in order to estimate the probability of a certain linguistic class occurring with a certain linguistic context.

$$S(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

2.1.2 Support Vector Machines

Support Vector Machines (SVM) are supervised learning models often used for text categorization and sentiment analysis, where the problem is either a classification or regression task. In the problem of classification, the goal is to find a hyperplane that differentiates the classes by having the largest distance to the nearest training data-point of each class. Training data instances are represented as coordinates in an n-dimensional space, where n equals the number of features. The hyperplane is a subspace with n-1 dimension, i.e. one dimension less than the number of features. The data points closest to the hyperplane are called support vectors, and the distance from the support vectors to the hyperplane is called the margin. As the goal is to find the optimal hyperplane with the largest margin possible, this classifier is also known as the *maximum margin classifier*. Figure 2.1 illustrates linearly separable training data, along with labels for the most important SVM elements. When the classification problem is not linearly separable, the algorithm can use kernel functions. Kernel functions are used to transform low-dimensional input space into a higher dimensional space which is then linearly separable.

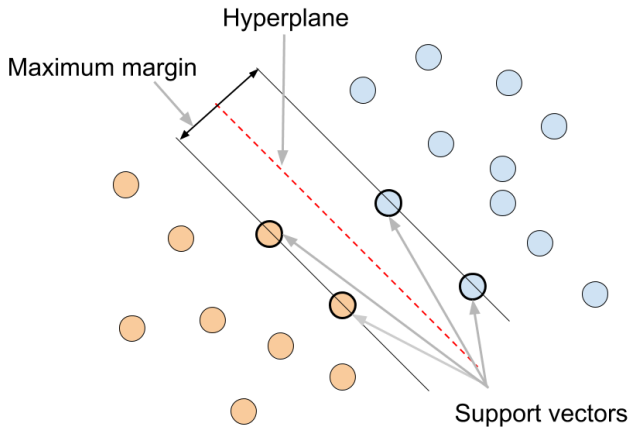


Figure 2.1: SVM hyperplane separation

2.1.3 Deep Learning

Deep learning is a subfield of machine learning that has seen considerable growth in popularity and usefulness in recent years. Goodfellow et al. (2016) describe two perspectives of deep learning; the first being to learn complex concepts out of simpler concepts, and the second that depth enables the computer to learn a multistep computer program. Artificial Neural Networks (ANNs) are networks inspired by the biological brain, and is one of the terms that deep learning has gone by. The first and simplest types of neural network are *feedforward neural networks*. Due to simplicity, these networks will be used to explain some basic concepts of neural networks before briefly presenting the ideas behind more advanced networks.

Feedforward networks are directed networks, where information only moves in one direction without cycles. In its most basic appearance its called a *single-layer perceptron*, consisting only of an input layer and an output layer. However, *multi-layer perceptrons* (MLP) also contain at least one hidden layer, and can learn both linear and non-linear functions. A simple multi-layer perceptron with one hidden layer and two output classes is illustrated in Figure 2.2. The basic unit of neural network are the *neurons*, which receive input and compute an output. Given

the inputs, the output is defined by the *activation function*, which can introduce non-linearity to the output. MLPs learn through the *backpropagation algorithm*. In simple terms, the backpropagation algorithm iteratively adjusts the weights in the network until the output is sufficiently correct. By using a loss function which measures the actual output against the predicted output, the error is "propagated" back to the previous layer and the weights are adjusted accordingly. To find the weights that minimize the outcome of the loss function, the *gradient descent* algorithm is used. Gradient descent is an optimization algorithm that iteratively moves towards minimizing a function, and is applied in several machine learning methods where parameters cannot be directly calculated.

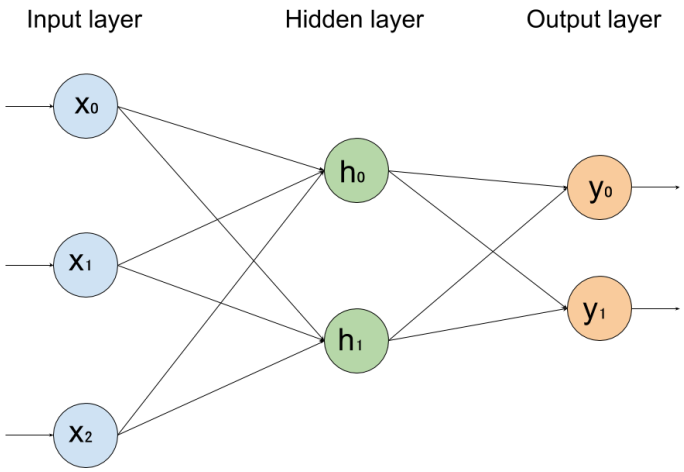


Figure 2.2: Feedforward network architecture

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are feedforward networks that have proven to be very successful in image recognition and classification, but have also been used in NLP tasks. Goodfellow et al. (2016) explain how typical CNNs consist of the convolutional layer, a non-linear, and a pooling layer, in addition to the input and output layer. In the first stage linear convolutional functions are applied, to

extract features from the input. Then, a non-linear activation function is applied which introduces non-linearity to the network. Lastly, a pooling function reduces the feature dimensionality while retaining the most important information. This modification helps to make the feature representation invariant to small translations of the input. CNNs attempt to find and learn the most relevant patterns of how to accomplish a given task while storing few parameters, which can reduce the memory required and improve model efficiency.

Recurrent Neural Networks

Recurrent Neural Networks (RNNs) have shown great promise in NLP tasks. As opposed to MLPs and CNNs that are directed and assume that inputs and outputs are independent, RNNs are cyclic and make use of sequential information (Goodfellow et al., 2016). Thus, RNNs are not feedforward networks. When RNNs perform tasks for elements in a sequence, the output is dependent on the previous computations. This is why the networks are called *recurrent*. It is said that RNNs have a "memory" that stores information about previous calculations. Figure 2.3 illustrates a simple RNN with one hidden layer and two output classes, where the black boxes in the hidden layer represent the network memory.

Long Short-Term Memory Neural Networks

Long Short-Term Memory (LSTM) networks are special types of RNNs that can address the shortcomings of regular RNNs. In their study, Bengio et al. (1994) showed how regular RNNs using gradient descent performed poorly for tasks involving long-term dependencies, and the LSTM architecture, introduced by Hochreiter and Schmidhuber (1997), was explicitly designed to overcome this. LSTM networks consist of units called *memory cells*. A memory cell has three gate units, with the ability to remove or add information to the memory cell state, and store temporal information. The *input gate* protects the memory state from being disturbed by irrelevant information and the *output gate* avoids storing irrelevant

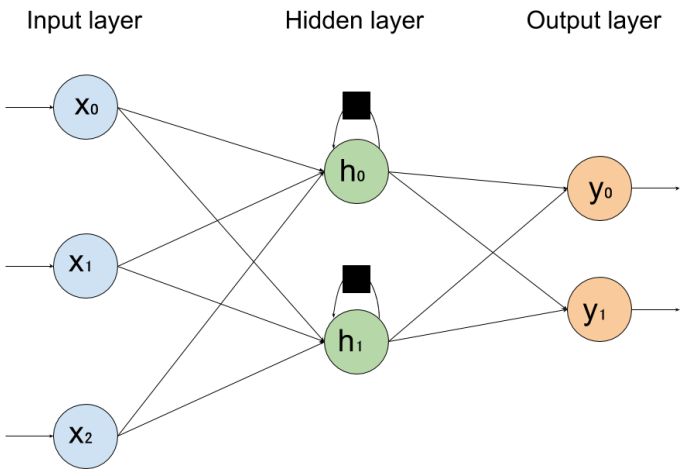


Figure 2.3: RNN architecture

information in the memory state. The *forget gate* controls which information to forget or store for later use. LSTM networks are particularly successful in solving tasks where capturing long-term and temporal dependencies is essential.

2.2 Evaluation Metrics

This section presents metrics that are often used in evaluation of classification model performance. These measures use the values of true positives, true negatives, false positives and false negatives. True positives (tp) denote the number of correctly classified positive instances, while true negatives (tn) denote the number of correctly classified negative instances. False positives (fp) denote the number of incorrectly classified positive instances, while false negatives (fn) are the same for negative instances.

2.2.1 Precision

The precision metric is the fraction of relevant instances among the retrieved instances. Intuitively, precision measures the ability of the classifier to correctly label samples. The formula for precision is given as:

$$precision = \frac{tp}{tp + fp} \tag{2.2}$$

2.2.2 Recall

The recall metric is the fraction of relevant instances that have been retrieved among all relevant instances. Intuitively, recall measures the ability of the classifier to find all the relevant samples. The formula for recall is given as:

$$recall = \frac{tp}{tp + fn} \tag{2.3}$$

2.2.3 F_1-score

The F_1-score is a harmonic mean of the precision and recall values, and is used for a better overall evaluation of the classifier performance.

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2.4}$$

There are different ways of calculating the average F_1-score, precision and recall. Two commonly used calculations include the “macro average” and “micro average”. Micro averaging calculates the metrics globally by counting the total true positives, false negatives and false positives. Macro averaging calculates the metrics for each label and finds their unweighted mean without considering imbalance of labels.

2.3 Natural Language Processing

Ambiguous and unstructured language is difficult for computer systems to interpret and process. Natural Language Processing (NLP) is a field of computer science that offers methods for enabling computer systems to make sense of natural language text. This chapter will briefly introduce some of the methods and concepts within textual preprocessing, representation and common features.

2.3.1 Textual Preprocessing

Preprocessing is an important step of NLP, especially when handling user generated content posted online. Preprocessing can be considered as a step for removing noise, and is done for a more easily extraction of features and information. When it comes to online writing, people often disregard grammar rules, misspell words, and use abbreviations. For a machine to be able to understand and make sense of such human-written language, the preprocessing step becomes essential. A common preprocessing pipeline often consists of the steps tokenization, stop-word removal, and stemming or lemmatization. Tokenization includes transforming the raw text into separated units, or tokens, and removing punctuation or special characters. In the process of stop-word removal, frequent words that often contribute little semantic value to a sentence are removed to reduce the vocabulary size and for more efficient search. Examples of common stop-words include “a” and “the”. Stemming reduces words to their word stem, such as reducing “dresses” to “dress”, while lemmatization converts words to their base form, such as converting “exploring” to “explore”. The main difference is that lemmatization requires a dictionary with a set of word base forms, while stemming operates based on rules. Other preprocessing methods also include spell-checking and removal of special characters, such as emoticons. Examples of such methods will be presented in Chapter 3.2.2.

2.3.2 Text Representation

Yan (2009) describes text representation as one of the fundamental problems in text mining that aims to numerically represent unstructured text to be mathematically computable. There are several representation models that have been proposed for NLP tasks and Information Retrieval (IR). This section introduces a few of these models, along with their advantages and limitations.

Term Frequency-Inverse Document Frequency

Term frequency-inverse document frequency (TF-IDF) is a statistical approach that assigns weights to terms according to their importance to a document in a corpus. The intuition is that the more times a word appears in a document, the more important is this word to the document. On the other hand, if a word appears in several documents in a corpus, the discriminative power of the word becomes weak (Sparck Jones, 1972). The former is represented by the term frequency (TF) value, and the latter by the inverse document frequency (IDF) shown in Equation 2.5. A term receives a high TF-IDF value, as calculated with Equation 2.6, if it appears in relatively few documents, but several times within those that it appears. TF-IDF weights are simple to compute and useful for stop-word removal and calculating similarities between documents. However, TF-IDF suffers the limitation that it cannot register relationships between terms or compute semantics.

$$idf = log\frac{N}{df_t} \tag{2.5}$$

$$tf\text{-}idf = tf_{t,d} \times idf_t \tag{2.6}$$

Bag of Words

The Bag of Words (BoW) model is a commonly used and simple text representation model. The model contains a vocabulary of known words, as well as a measure of their presence. The simplest measure of presence is Boolean, either present in a document or not, while more advanced measures can include the frequency or TF-IDF values. Figure 2.4 shows a bag of words representation of two documents with the text "The dog likes to swim" and "John likes to swim with his dog". The measurement used for the words' presence is the frequency of each word's appearance in the documents. Le and Mikolov (2014) express that the main limitation of the BoW model is that it ignores the order of the words, and that different sentences may have exactly the same representation if the same words are used. In addition, the model does not consider the semantics or distances between words.

- **Document 1**: The dog likes to swim in the pool
- **Document 2**: John likes to swim with his dog

	the	dog	likes	to	swim	John	with	his	in	pool
D1	2	1	1	1	1	0	0	0	1	1
D2	0	1	1	1	1	1	1	1	0	0

Figure 2.4: BoW representation

N-grams

N-grams is a statistical language model consisting of a sequence of items, either characters or words, where n denotes the length of the sequence. When n=1, the model is called a unigram, and can be considered a special type of the BoW model. For n = 2, the model is called a bigram, for n=3 a trigram, and so on. N-grams address the limitation of the BoW model by including word order, for lengths $n > 1$. N-grams are highly popular in NLP tasks and can have several applications, such as finding likely candidates for misspelled words, predict the next word in a sequence, or capture term correlation. A limitation of the n-gram model is that the representation may not be able to capture long-distance dependencies between

words. An example of word n-gram representations of the text "The dog likes to swim" is shown in Figure 2.5.

- Unigram (1-gram):

The	dog	likes	to	swim

- Bigram (2-gram):

The dog	dog likes	likes to	to swim

- Trigram (3-gram):

The dog likes	dog likes to	likes to swim

Figure 2.5: Word n-gram representation

Word Embeddings

Turian et al. (2010) define word embeddings as distributed word representations where each dimension of the embedding represents a latent feature of the word, hopefully capturing useful syntactic and semantic properties. Word embeddings have become highly popular, and are learned from the usage of words to create a dense representation where words that are used similarly also have similar representations. Examples of models for constructing word embeddings from text include *word2vec* and *GloVe*, and these will be further explained due to their popularity. Word2vec is a predictive algorithm developed by Mikolov et al. (2013), and consists of the two models Continuous Bag-of-Words (CBOW) and Continuous Skip-Gram. The CBOW model aims to predict a word from a window of surrounding words, while the skip-gram architecture uses a current word to predict a window of surrounding words. The GloVe algorithm serves as an extention to the word2vec algorithm, developed by Pennington et al. (2014). GloVe is an unsupervised learning algorithm that uses statistics to produce a word vector space with meaningful substructure. This model outperforms other models on word analogy, word similarity and named entity recognition tasks.

2.3.3 Features for Hate Speech Detection

When the amount of training data is large enough, the performance of different classification methods becomes more similar. The distinguishing impact on performance will then come from the features chosen to employ in the methods. This section describes common types of features used in NLP, based on a survey of existing research on features used in hate speech detection (Schmidt and Wiegland, 2017).

Simple Surface Features

The presence and frequency of the words in a document are simple features that are easily retrievable. These features can be derived without advanced methods, and include bag-of-words, n-grams, and appearances and frequencies (URL, words, characters, etc). Nobata et al. (2016) found that n-gram features are predictive and perform well on their own in noisy data sets, and that a combination with other features is shown to be powerful.

Word Generalization

The problem of data sparsity in text representation can be approached by applying a form of word generalization, such as word embeddings or word clustering. In word clustering, each cluster with a set of words can be used as a feature. Words can either be fully assigned to a particular cluster or can be assigned a degree of belonging to each cluster. Word embeddings can be considered both as word representations and features. As features, embeddings may replace the presence or frequency of particular words, by instead establishing the similarities between words in the representations.

Lexical Resources

Lexical resources, or word lists, are necessary when the presence of specific words are used as features. There are several word lists available on the Internet for general tasks, and some are also made publicly available. For very specific NLP tasks, it may be convenient to create a new list or dictionary. Lexical features are considered insufficient as stand-alone features, and is recommended to be combined with other types.

Syntactic Features

Syntactic or linguistic features consider the structure of text and relationship between words, enabling a better understanding of the underlying meaning. Part-of-speech tagging is a method for marking words in a text according to their respective part of speech, such as nouns, verbs or adjectives. The challenge is assigning part-of-speech tags to words that can have different meanings. Dependency relationships are employed to capture relationships and dependencies between words, which is useful for capturing relations between non-consecutive terms.

Context-Based Features

The textual context can provide useful information in understanding the meaning and opinions in a text. However, the context may be difficult to both retrieve and represent. Knowledge-based features are useful in specific domains, but require a creation of the knowledge-base, which can be comprehensive to implement. Meta-information provides information that is not directly present in the text but can be derived from the surroundings, such as information about the author or an article that is referenced to. Online posts are often a combination of multiple modalities, such as text, video and image. Then, methods beyond those of NLP may have to be applied to fully understand the context. Features such as hashtags or image labels or categorizations can be useful to employ in these situations.

Sentiment Analysis

Sentiment analysis is concerned with identifying the sentiment or subjective content of a text. Common features used for sentiment identification include the presence of positive or negative words, the use of emoticons, or word dependencies. There exists several tools that assist in detecting sentiment or polarity, that are commonly used in NLP tasks where sentiment analysis is essential.

2.4 Tools

This section provides a description of the tools and libraries that were used in this thesis.

2.4.1 Twitter API and Tweepy

Twitter provides various Application Program Interfaces (APIs) for developers to engage with the Twitter platform. The Twitter REST API allows developers to access Twitter data, such as information about users, timelines, and tweets. Tweepy is a Python library used for accessing the Twitter API, and specifically the objects and methods that the API offers. The Tweepy library was used to retrieve the different user information to be used in the data analysis and in the experiments.

2.4.2 Natural Language Toolkit (NLTK)

The Natural Language Toolkit (NLTK) by Bird et al. (2009) is a platform for natural language processing using the Python programming language. NLTK offers interfaces to corpora and lexical resources, and libraries for common NLP techniques such as tokenization, stemming and parsing. In the text preprocessing of

the tweets from the datasets, NLTK was used for tokenizing, stop-word removal, lowercasing and removal of special characters.

2.4.3 Scikit-learn (sklearn)

Scikit-learn (sklearn) in an open source Python library for machine learning. Sklearn was initially developed by Pedregosa et al. (2011), and features support for several supervised and unsupervised machine learning algorithms. The sklearn library offers several modules that were used to implement the baseline hate speech classifier used in the experiments of this thesis. These modules are described below.

Transformers

Scikit provides a library of transformers to transform data, such as preprocessing, dimensionality reduction or generating feature representations. Transformers were used to generate feature representations of the tweets, and to scale the numeric user features.

Pipeline

The pipeline module allows creating a single object that includes all steps from data preprocessing to classification. Specifically, a pipeline allows for a convenient chaining of a sequence of estimators, where the final estimator is often a machine learning classifier. Pipelines also provide support for cross validation and grid search of the parameters of all estimators in the pipeline at once.

Feature Union

Feature unions can combine several feature extraction methods into a single transformer, and are therefore useful for datasets that consist of heterogeneous data

types, e.g. images and text. It is common to include a feature union as a step in a pipeline.

Grid Search

With a specified hyperparameter space, grid search functions can exhaustively search the space for parameters that result in the best model performance. Grid search was used to find the best type and range of n-gram features for the implemented hate speech classifier, and will be explained in detail in Section 5.2.

3 Related Work

This chapter first presents a review of the existing research in the field of hate speech detection, as well as the challenges faced in the field. Then, the state-of-the-art approaches within the field are introduced, and lastly the chapter presents studies focusing on the authors of hate speech, which is particularly relevant for this thesis.

3.1 Studies on Hate Speech Detection

The societal impact of the Internet and social media has increased over the past years, and perhaps this is why there has also been a growth and interest in research covering hate speech detection. While the amount of research increases, the field still faces several challenges, both in the actual task of detecting hate speech and the research area in general. Nobata et al. (2016) have summarized the following challenges for the task of detecting hate speech. First, what is interpreted as offensive or hate speech is subjective, and can differ from person to person. This can be a problem for annotation of data for training hate speech detection systems, as the annotators do not necessarily agree. Ross et al. (2016) aimed to estimate the reliability of annotations, and found that there is a low agreement among users when it comes to identifying hateful messages and that hate speech requires a significantly better definition and guidelines in order to be annotated reliably. Second, the language that is used for expressing hate speech is constantly evolving, with users introducing new terms and methods to avoid being detected and with language that follows trends or specific incidents. Hate speech detection

is also more than simply spotting bad words. While much of offensive language contains noise and offensive words, hate speech can also be very fluently written and grammatically correct. Lastly, hate speech can also be a result of several sentences or even posts, making it more difficult to detect. Ventirozos et al. (2017) considered the whole message thread in their method for detecting hate speech. By doing this, the authors were able to extract features that relate to changes in sentiment between consecutive messages, which single messages cannot provide.

There are also several challenges concerning the actual research field. The lack of a benchmark dataset makes it difficult to compare studies and methods. In addition, there is not a common understanding of the task or terms. Although there is a common and overall goal of detecting hate speech, there are variations in how this goal is approached and the subtasks that have been studied. Kwok and Wang (2013) and Waseem and Hovy (2016) have focused on detecting racism in Tweets, and the latter also released a dataset containing racist and sexist language. This dataset has been widely used in other research. Many researchers have focused their studies on detecting profanity or offensive language. Sood et al. (2012b) used crowdsourcing to improve profanity detection, which outperforms the list-based approach previously used. Some studies also focus on determining whether or not users or comments will be moderated by their online community. Davidson et al. (2017) differentiated the degree of hate speech, and classified instances as either hate speech or offensive language. Although it may be considered as a different task than hate speech detection, several studies have also focused on detecting cyberbullying, personal attacks and trolling.

As the amount of work on hate speech detection increases, some studies aim to create overviews of existing work within the field of study, or specifying definitions. The overview developed by Schmidt and Wiegland (2017) serves as an introduction to researchers that are new to the field of hate speech detection and want information about the state-of-the-art, and feature extraction in particular. Fortuna (2017) dedicated her master's thesis to creating a complete overview of what she considers a field of study in an early stage with many opportunities. Her thesis also contributes an annotated dataset in Portuguese, along with a review

of existing datasets and annotation methods in the research field. As mentioned, there does not exist one formal definition of hate speech, nor a definition of the task of detecting it. Motivated by the need to clarify the relationships between different subtasks of abusive language detection, Waseem et al. (2017) have created a typology of abusive language, hoping to clarify the key aspects of abusive language detection. The typology proposes that subtasks of hate speech can be categorized into being either generalized or specific, and explicit or implicit. The authors encourage future researchers to learn from advances in related areas, to use the appropriate features for each subtask, to take into account that not all abuse is equal, and to be more transparent in discussing the annotations and modeling strategies used.

3.2 State-of-the-Art

This section describes the state-of-the-art within hate speech detection. First, the issues concerning datasets will be presented along with the most commonly used datasets for the task. Then, the methods within preprocessing, feature extraction and classification methods will be introduced in order.

3.2.1 Datasets

The lack of a benchmark dataset for the task of hate speech detection is an issue as it becomes difficult to compare methods and results that are based on different data and annotations. In addition, the datasets are created for different tasks, and therefore have different characteristics and display different types of hate speech. Creating datasets for this task is time consuming, as the number of hateful instances in online communities is relatively few, but it is necessary to have a representable amount of such instances in a dataset. There are also several datasets that have not been made publicly available. This may be due to privacy issues or considering the content of the datasets, i.e. the profanity and offensive

language. The authors behind the Gazzetta dataset (Pavlopoulos et al., 2017a) used an encryptor on their dataset before publishing it online. Altough this is a simple encryptor, the intention was to avoid direct hate speech content on the Internet. Despite these challenges, there has been an increase of datasets created for the task with the contribution of making them publicly available and in other languages than English. Table 3.1 provides an overview of the known datasets, and datasets used in the related works of this thesis.

3.2.2 Preprocessing

In the task of hate speech detection, common preprocessing steps include tokenization, stop-word removal and stemming. These are simple steps that are applicable in all NLP domains and contribute to reducing the size and variations in a document, and might thereby improve the efficiency and effectiveness of the information retrieval. Sood et al. (2012a) used stemming as an attempt to improve recall in comparison to simple list-based approaches for detecting profanity. In their approach, the stemmer allowed the system to see if there were any words in a comment that shared the same stem with any words on a profanity list. The stemmer approach combined with a profanity list showed better performance than the list-based approach; however, the overall performance was poor and the system only detected 40.2% of the profanity cases. The authors concluded that profanity detection is a difficult task due to the evolvement of profane language, the intentional and unintentional misspellings and the authors' effort in disguising profane words. Although these simple preprocessing methods contribute to making the raw text more readable and easier to work with, it might often be necessary to process the text in even more advanced matters. As a part of their preprocessing, Papegnies et al. (2017) mapped hexadecimal or binary encoded text back to ASCII, as it is seen that this is an option for disguising profanity.

Other common methods for preprocessing include lowercasing the raw text and removing special characters or punctuation. Burnap and Williams (2015) transformed all tokens in tweets to lowercase to avoid capitalized and lowercased versions

Authors	Source	Size	Annotations	Language	Available
Waseem and Hovy (2016)	Twitter	16K tweets	Racist, sexist	English	Yes
Wulczyn et al. (2017)	Wikipedia	115K comments	Personal attack	English	Yes
Davidson et al. (2017)	Twitter	24K tweets	Hate speech, offensive	English	Yes [a]
Nobata et al. (2016)	Yahoo	2,1M comments	Abusive	English	No
Pavlopoulos et al. (2017a)	Gazzetta	1,6M comments	Moderated	Greek	Yes
Djuric et al. (2015)	Yahoo	950K comments	Hate speech	English	No
Fortuna (2017)	Twitter	5,668 tweets	Hate speech	Portuguese	Yes
Papegnies et al. (2017)	SpaceOrigin	2,337 messages	Abusive	French	No
Gao and Huang (2017)	Fox News	1,528 comments	Hateful	English	Yes
Ross et al. (2016)	Twitter	541 threads	Hate speech	German	Yes
Golbeck et al. (2017)	Twitter	35K tweets	Online harassment	English[b]	Yes[a]
Chatzakou et al. (2017)	Twitter	9,484 tweets	Aggressive or bullying	English[b]	No

Table 3.1: Overview of existing datasets for hate speech detection

[a] Available upon request
[b] Not explicitly stated

of words being treated as separate words. This approach resulted in a more dense representation of the vocabulary, which can more easily capture synonymity than sparse representations. Chen et al. (2012) mentioned that users expressing themselves online may use punctuation and words with all uppercase letters to indicate feelings or speaking volume. Thus, lowercasing and removing special characters in a text may result in some features being missed. In their own research, Chen et al. (2012) used a spell-correction algorithm in their preprocessing. Their algorithm corrected spelling and grammar mistakes by using tasks such as deleting repeated letters in words, transposing substituted letters, splitting long words, and replacing incorrect and missing letters in words. Xiang et al. (2012) designed a word cleaning algorithm based on the same tasks, and this type of spell-checking has become a common part of preprocessing in many works. Papegnies et al. (2017) pointed out that the tendency to misspell words can be an important feature to describe the user, and therefore preprocessing should be applied with caution as blind preprocessing would hide that feature. Nobata et al. (2016) share this view about spell-checking and normalizing text before feature extraction. In their study, noise in a text is considered as a good signal for abuse detection, and the authors employed features in their methods to capture different types of noise.

3.2.3 Feature Extraction and Representation

After the text has been preprocessed, features that will provide information and reflect its class are extracted. The features must also be represented in a suitable manner before being fed into a machine learning classifier. While Section 2.3.3 described the different types of features, this section presents the state-of-art within feature extraction and representation in hate speech detection.

Lexical and syntactic features have been the most commonly extracted features in hate speech detection. Lexical features are employed by many researchers, and the use of appearance and frequency of bad words is very common. This may be based on the assumption that hate speech often contains bad words. Kwok and Wang (2013) found that 83% of their tweets were annotated racist because they

contained offensive words, and were therefore motivated to use unigram features when constructing their vocabulary. However, only employing unigrams made it difficult to capture relationships between words, and the system mistakenly classified tweets containing offensive words as racist, thereby reducing accuracy. To avoid this, Nobata et al. (2016) used syntactic features with the motivation being that syntactic features capture long-distance dependencies between words which n-grams may not be able to. Syntactical features can be derived from analysis of the actual text and are therefore highly available. Other features employed by Nobata et al. (2016) include n-gram features, several linguistic features and distributional semantic features or embedding derived features. In their method to detect abusive language and evaluate the performance of several NLP features for this task, they found that combining all features yields the best performance. However, in terms of individual features, character n-grams proved to make the largest contribution and performed well in the noisy datasets used.

Other studies also have the common understanding that character n-grams provide the best contribution. Waseem and Hovy (2016) investigated which of the features used in their method provided the best identification performance, and the study found that using a character n-gram based approach provided a solid foundation. Their study also analyzed the impact of extra-linguistic features, such as gender and location. The results indicated that demographic information, apart from gender, brings little improvement. However, this could be due to lack of coverage. In their study, Mehdad and Tetreault (2016) specifically investigated character-based features, and compared them to token-based features. The findings showed that character-based approaches are superior to token-based approaches and other state-of-the-art methods. The motivation for using character n-grams was based on the observation that user language evolves due to the standards and guidelines that users of online communities must follow. Users learn over time how to avoid blacklisted words and disguise certain language, and therefore the use of characters-based methods play an important role in hate speech detection as these methods can provide a way to deal with spelling variations.

Distributional semantic features, or embedding derived features have also been em-

ployed in several methods with promising results. In their method for hate speech detection, Djuric et al. (2015) learned distributed low-dimensional text embedding of comments using the continuous BOW neural language model, where semantically similar comments and words lie in the same part of the space. The embeddings were then used to train a logistic regression classifier to detect instances of hate speech. This approach addressed the issue of high-dimentionality and sparsity that often affects the methods that use bag of words (BOW) representations. The findings indicated that the proposed approach was highly efficient and effective in hate speech detection, and outperformed the BOW methods in the study. In a different approach, Pavlopoulos et al. (2017b) added user embeddings or user type embeddings to a method for abusive comment moderation. While word embeddings represent words, user embeddings are dense vectors that represent individual users or user types. In a state-of-the art RNN method, adding user embeddings, user type embeddings, user biases or user type biases resulted in improvements. However, the addition of user embeddings resulted in the largest improvement.

Feature selection is the problem of choosing the features that are most useful and best represent the underlying problem. Robinson et al. (2018) recently conducted a feature selection analysis using Twitter data for hate speech detection. By using surface features, linguistic features and sentiment features in an SVM-based hate speech classifier, they find that automatic feature selection can significantly reduce carefully engineered features by over 90%. Furthermore, the study found that feature selection resulting in a small set of predictive features achieves much better results than models using carefully engineered features.

3.2.4 Classification Methods

Supervised machine learning classifiers have been the most frequently used approaches for the task of online hate speech detection. Support Vector Machines (SVM), as presented in Chapter 2, are popular classifiers used in several tasks in NLP, and are perhaps the most common classifiers in hate speech detection, as stated by Schmidt and Wiegland (2017). Logistic regression (LR) is also a pop-

ular choice, imaginably due to its simplicity. Davidson et al. (2017) first used a logistic regression model to reduce the data dimensionality before testing a variety of classification models. In their study, it was found that the logistic regression and linear SVMs tended to perform better than the other models, such as naïve bayes, decision trees and random forests. However, the best performing model still misclassified almost 40% of the hate speech instances in the dataset, implying that automated hate speech identification is a difficult task.

As mentioned in Chapter 2, there has been a recent growth in the use of deep learning methods for machine learning tasks. This is also the case for the specific task of hate speech detection. Pavlopoulos et al. (2017a) explored how deep learning can be used to moderate user comments. By comparing different types of Recurrent Neural Networks (RNN), a Convolutional Neural Network (CNN) and the system developed by Pavlopoulos et al. (2017a), the study found that an RNN operating on word-embeddings provided the best results for abusive comment moderation. This means that the RNN outperformed the previous state-of-art method that used an LR or MLP classifier with character and word n-gram features. Mehdad and Tetreault (2016) also exploit RNNs, with the purpose to overcome the challenge of learning with little training data. Badjatiya et al. (2017) experimented with various deep learning architectures to learn semantic words embeddings in an attempt to handle the complexity of natural language. Their experiments showed that these deep learning methods outperform character and word n-grams, which was previously considered the state-of-the-art. Several authors express the need for more experimenting with neural networks, and Wulczyn et al. (2017) particularly mentioned their desire to experiment further with Long Short-Term Memory (LSTM) networks.

While there are several studies that have compared the performance of different classification methods, there are still no studies that can determine the most effective approach to hate speech detection. A comparative study performed by Burnap and Williams (2015) concluded that an ensemble method seemed most promising. However, these results might have been heavily affected by the feature set or dataset, and the method can therefore not be acknowledged as the ideal

approach. Zhang et al. (2018a) implemented a new method based on convolutional and recurrent networks, that was evaluated against several other baseline and state-of-the-art methods on different datasets. The study found that the new method outperformed the baselines methods on 6 out of 7 datasets. This again shows that deep neural network models are very promising for the task of hate speech detection.

3.3 Authors of Hate Speech

Related to the studies on hate speech detection is the studies of the people that post hateful content online. These studies outline characteristics and behavioral traits that are typical of the authors behind aggressive behavior, hate speech or trolling. The motives for these people can differ, and where some people want to reach out to a large audience to create a negative atmosphere, some target specific individuals. The findings from these studies can contribute to indicating which information about a user could be useful to employ in hate speech detection methods.

To better understand the nature of personal attackers, Wulczyn et al. (2017) qualitatively analyzed a large annotated corpus containing personal attacks. Several questions were investigated, including: What is the impact of anonymity? How do attacks vary with the quantity of a user's contribution? Are attacks concentrated among a few highly toxic users? When do attacks result in a moderator action? And is there a pattern to the timing of personal attacks? This research based on Wikipedia comments revealed that anonymity increases the likelihood of a comment being an attack, but anonymous comments only contribute to less than half of the total attacks. About 30% of the attacks come from registered users with over a 100 contributions, and less than half of the attacks come from users with little participation. The study also suggested that personal attacks cluster in time, which may be because one attack triggers another. In another qualitative analysis, Cheng et al. (2015) characterized forms of antisocial behavior in online discussion communities. The study compared the activity of users that are permanently

banned from a community (*Future-Banned Users (FBUs)*) and users that are not banned (*Never-Banned Users (NBUs)*). The study found that FBUs tend to write less similarly to other users, use less positive words and more profanity, and these users tend to concentrate their efforts in a small amount of threads. FBUs also receive more replies and responses than other users. In a longitudinal analysis the study also found that behavior of FBUs worsen if they are excessively censored early in their lives.

Hardaker (2010) extracted an academic definition of *trolling* from user comments in her study. The study suggests the following definition: A troller is a computer-mediated communication (CMC) user who constructs the identify of sincerely wishing to be part of the group in question, including professing, or conveying pseudo-sincere intentions, but whose real intention(s) is/are to cause disruption and/or to trigger or exacerbate conflict for the purposes of their own amusement. In another study about internet trolls, Buckels et al. (2014) aimed to find their characteristic traits. By conducting a study on commenting styles and personality inventories, the authors found strong positive relations among commenting frequency, trolling enjoyment and trolling behavior and identity. The research also concluded that of all included personality measures, sadism had the most robust association with trolling. Cheng et al. (2017) proposed that an individual's mood and seeing troll posts by others are trigger mechanisms for engaging troll behavior. Specifically, by simulating an online discussion and longitudinal analysis of a discussion community, the authors found that mood and discussion context together better explain trolling behavior than an individual's history. The results also suggested that ordinary people can engage in such behavior as well, under specific circumstances.

Several studies have expressed the need for, and the importance of, including user information in methods for detecting hate speech, and this is still considered an under researched area. Lynn et al. (2017) discussed the concept of human centered NLP, and recognized their work as a part of a growing trend to put language beyond its document-wide context, and also within the context of its authors. Although this trend is seen in the general field of NLP, it is highly relevant in hate

speech detection as well. On the more technical side, Chen et al. (2012) proposed a new architecture, the Lexical Syntactic Feature (LFS) architecture, to bridge the gap between detecting offensive content and potential offensive users in social media. The motivation behind the LFS architecture was that existing methods treat messages as independent instances, whereas they should focus on the source of the offensive content to improve the offensive content detection. Papegnies et al. (2017) plan to propose other context-based features for abuse detection, and especially those based on the networks of user interactions. Several authors share this intention, but face the challenge that information about the users is often unavailable, or very limited.

Chatzakou et al. (2017) presented an approach specifically to detect and label bullying and aggressive behavior of users on Twitter. Similarly to the objective of this thesis, their study investigated user features that can be used to enhance classification algorithms. More specifically the study aimed to distinguish bullies and aggressors from regular users. Their study found that network-based features were particularly useful and effective in classifying aggressive user behavior. Network-based features included metrics such as the number of friends and followers, reciprocity and the position in the network. Another contribution of their study was the developed dataset containing tweets from users labeled as normal, spammer, aggressor or bully. The findings from Chatzakou et al. (2017) are related to the topic of this thesis, and will therefore be useful for comparison of the experimental results.

Table 3.2 provides an overview of the characteristics and users features from the related studies presented mainly in this section. The table presents highlighted features or information, an explanation of each, and the authors supporting the findings.

Feature	Explanation	Supported by
Commenting frequency	Trolling enjoyment and troll identity have strong positive associations with online commenting frequency	Buckels et al. (2014)
User activity	Users who have been banned from an online community post more frequently than users who are not banned	Cheng et al. (2015)
Thread activity	Users who have been banned from online communities often concentrate on few threads rather than many	Cheng et al. (2015)
Activity level	Users at both low and high activity levels (number of comments) of contribution are responsible for a significant portion of attack	Wulczyn et al. (2017)
Moderation	The likelihood of a new attack leading to a block increases with the number of times a user has been blocked in the past	Wulczyn et al. (2017)
Anonymity	The personal attack prevalence among comments by anonymous users is higher than the prevalence of registered users	Wulczyn et al. (2017)
Surrounding context	Seeing troll posts by others increases the probability of a user trolling	Cheng et al. (2017)
Clustering	There is a strong indication that personal attacks cluster in time on Wikipedia discussions	Wulczyn et al. (2017)
Response	Users who have been banned from an online communities users receive more response than other users	Cheng et al. (2015)
Network	Network-based attributes are very effective features for detecting aggressive user behavior	Chatzakou et al. (2017)
Gender	In a set of various extra-linguistic features, only gender brings improvement to the hate speech detection system	Waseem and Hovy (2016)
Mood	An individual's negative mood can cause trolling behavior	Cheng et al. (2017)

Table 3.2: Overview user characteristics and features from related work

4 Data

Machine learning methods learn functions from data. While the representation and amount of data needed depend on the complexity of the problem to be solved, it is beneficial that the data is somewhat representative of the related real-world problem. Creating labeled datasets can be a tedious and demanding task. Therefore, already existing datasets will be used in this thesis. The data from the datasets is used to investigate characteristics of Twitter users, and then to train and evaluate the classification model for hate speech detection, where the latter is described in Chapter 5. Section 4.1 presents the datasets used and their distribution of labeled instances. Available information about the users was extracted from the data and used to analyze different characteristics of the datasets, which are presented in Section 4.2.

4.1 Datasets

Several datasets were used to investigate the characteristics of users for increased insight and to allow comparisons of the findings. All datasets have used Twitter as their source for data collection, which ensured that the same information could be retrieved. However, the datasets are different in terms of languages and annotations, which will be described in the next sections. The datasets contain tweet ID's which are unique integer representations of a tweet, that can be used to retrieve the actual tweeted text, information about the tweet or information about the user that has posted it. As user information is something that should be handled with care, it is important to mention that there has not been any

attempt to directly identify the users in the datasets. Tweet ID's may become unavailable, most probably due to the tweet having been deleted, that the user who posted the tweet has become suspended, or has deleted their account. Therefore, a review of the availability of the tweets in all datasets has been conducted prior to the investigation of characteristics. The analysis and experiments performed in this thesis will be based on the updated datasets.

4.1.1 Waseem and Hovy (2016)

One of the datasets used in this thesis was provided by Waseem and Hovy (2016), and has been made publicly available on GitHub[1]. The public Twitter search API was used to collect the corpus from Twitter, and in total 16,907 English tweets were annotated either as racist, sexist or neither. A majority of the tweets collected are related to an Australian TV show, and others were collected using a manual search of common terms used about religious, sexual, gender and ethnic minorities. Table 4.1 provides an overview of the original distribution and number of tweets and users of the dataset. The dataset contains more instances of neutral tweets than tweets with racist or sexist content. This unbalancing was intended by the developers, as the dataset becomes more representative of the real world problem where hate speech is a limited phenomenon. The inter-annotator agreement for this dataset was $\kappa = 0.85$, mainly caused by the lack of contextual information and different opinions of what determines sexism.

Considering that the dataset was developed in 2016, Tweepy was used to filter out any unavailable tweets and users. The results of the filtration are presented in Table 4.2. It was found that several of the tweets in the original dataset by Waseem and Hovy (2016) were unavailable, which impacted both the number of tweets and the number of users in the dataset. In addition to the filtering, the "Sexism" and "Racism" target classes were merged into one "Hate speech" class.

[1] https://github.com/ZeerakW/hatespeech

[2] An email (Z. Waseem, personal communication, February 10, 2018) confirmed this as the correct number, and not the number presented in the original article.

Label	Number of tweets	Number of users
Sexism	3,378	613
Racism	1,970	9
None	11,559	1,777[2]
Total	16,907	2,399

Table 4.1: Overview of original dataset by Waseem and Hovy (2016)

Label	Unavailable tweets	Available tweets	Number of users
Hate speech	341	5,007	539
None	663	10,896	1,569
Total	1,004	15,903	2,108

Table 4.2: Available tweets and users in dataset by Waseem and Hovy (2016)

4.1.2 Fortuna (2017)

Another dataset used for characteristics analysis and classification was provided by Fortuna (2017), created with the motivation to promote research of hate speech detection in Portugese. As a result, Fortuna (2017) developed a dataset consiting of 5,668 tweets in Portugese annotated with several instances of hate speech, available on INESCTEC [3]. Tweets were collected through the Twitter API with searches based on keywords related to hate speech and Twitter profiles known for posting hate messages. Only Portugese tweets were stored, and repetitive tweets and retweets were removed. Tweets that contained less than tree words, not counting hashtags, user mentions and URLs, were also removed. Lastly, a limit to only include a maximum of 200 tweets from one account was set to reduce the initial sample and ensure a more diverse source of tweets. The inter-annotator agreement was $\kappa = 0.72$. Table 4.3 shows the distribution of the annotated instances. The developer of the dataset aimed to have a higher proportion of hate speech messages than other datasets in the field of research, and in total 22% of the tweets were annotated as hate speech. In total there are 5,668 annotated tweets by 1,156 distinct users, however, the distribution of users within the target classes was not

[3]https://rdm.inesctec.pt/dataset/cs-2017-008

specified.

As with the dataset from Waseem and Hovy (2016), this dataset also has a considerable amount of unavailable tweets. Close to half of the tweets in both classes are unavailable, resulting in a total number of 3,059 available tweets. However, as shown in Table 4.4, there are still 1,010 users available, meaning that the unavailability of tweets did not heavily affect the number of users. While the original dataset had a binary value for the presence of hate speech and subcategories as labels, the target classes were here changed to "Hate speech" and "None".

Label	Number of tweets	Number of users
Hate speech	1,228	-
None	4,440	-
Total	5,668	1,156

Table 4.3: Overview of original dataset by Fortuna (2017)

Label	Unavailable tweets	Available tweets	Number of users
Hate speech	579	649	376
None	2,031	2,410	634
Total	2,610	3,059	1,010

Table 4.4: Available tweets and users in dataset by Fortuna (2017)

4.1.3 Ross et al. (2016)

To investigate the issue of reliability concerning hate speech annotation, Ross et al. (2016) compiled a German hate speech corpus with tweets linked to the refugee crisis in Europe. This is the third dataset used in this thesis. By using known hashtags that can be used in an insulting or offensive way, a total of 13,766 tweets were collected. A filtration was conducted by removing tweets containing a lot of non-textual content and only retaining original tweets that were understandable and linked to the refugee crisis. The corpus in total consists of 469 tweets, which were annotated by two annotators in order to determine if hate speech was present

or not. In addition, the offensiveness of a tweet was rated on a 6-point Likert scale. Ross et al. (2016) found that even with an outlined definition, the annotators had a low level of agreement (Krippendorff's $\alpha = .38$). Table 4.5 shows the distribution of the tweets, and the corresponding number of annotators who considered the tweet to contain hate speech. Table 4.6 shows the availability of the tweets in the dataset and the number of users in each target class. Similarly to the other datasets, a large number of tweets have become unavailable since the development of the dataset. It was beneficial to transform the labels of the dataset into binary classes, to equal the labeling of the other datasets. Therefore, a tweet that was labeled "Yes" by one or both of the annotators was assigned to the "Hate speech" class. In table 4.6, the available tweets in the "Hate speech" class consists of 65 instances labeled as hate speech by one annotator, and 33 instances labeled hate speech by both annotators.

Annotation	Number of tweets
Yes & Yes	54
Yes & No	100
No & No	315
Total	469

Table 4.5: Overview of available tweets in dataset by Ross et al. (2016). A "Yes" was given if one of the annotators meant that hate speech was present in the tweet

Label	Unavailable tweets	Available tweets	Number of users
Hate speech	56	98	47
None	72	243	123
Total	128	341	170

Table 4.6: Available tweets and users in dataset by Ross et al. (2016)

4.2 Characteristics

A quantitative analysis was conducted to better understand the characteristics of the users in the datasets, based on the proposed features in Table 3.2 and other information about the user available through the Twitter API. The research that supports the content in Table 3.2 is based on different types of hate speech, and the underlying assumption of this analysis is that the findings may also apply to the types of hate speech in the datasets used. This section presents the found characteristics and evaluates how the findings can be used as features to represent the underlying problem. The features from Table 3.2 that are not analyzed in this section are not included due to it not being possible to extract associated information through the Twitter API. All datasets have included several tweets from the same users, and therefore these users can be present in both target classes. To better distinguish between users and avoid redundancy in the analysis, users who are present in both target classes are only included as users within the "Hate speech" class and removed from the "None" class.

4.2.1 Gender

Twitter does not require users to register their gender, and therefore an explicit gender field is not retrievable through the Twitter API. Finding the gender distribution for users in the dataset is therefore challenging. In their study, Waseem and Hovy (2016) investigated the distributions of gender in the original dataset described in Section 4.1, and found that using character n-grams along with gender provided the best results in their experiments. The authors extracted gender information by looking up usernames and names in the user profiles, and then comparing these names to known male or female given names. A similar approach has been used in this thesis to find the gender distribution of the datasets, by using common international, Portuguese, German, and English names. In addition, the user description has also been considered, as this is a place where users often give a more detailed description of who they are, e.g "I am a mom of three boys". There may be a risk with this approach, as names or descriptions may mistakenly

be classified as the wrong gender, and therefore the gender findings may not be entirely accurate. Names that can be of both female and male gender have been avoided.

The gender distribution derived by Waseem and Hovy (2016) is illustrated in Figure 4.1. In their own study, Waseem and Hovy (2016) expressed that the gender of a considerable amount of users could not be identified by the used approach, and that the male gender was over-represented in all categories, and with a very low female representation. Figure 4.2 presents the gender distribution derived in this thesis, with significant differences to the findings in Figure 4.1. The main difference is that female users are identified to a much larger degree, and that the distribution of male, female and unidentified users appear more equal. In addition, the total percentage of unidentified users has decreased from 50% to 36%. The tendency of both figures show that a higher amount of male users are identified than female. In contrast, the gender distribution derived from the dataset by Fortuna (2017) in Figure 4.3 shows that a majority of identified user genders are actually female. Also in this dataset there is a large number of unidentified genders, with about 50% of all users in the dataset. Lastly, in the gender distribution of the dataset by Ross et al. (2016), as illustrated in Figure 4.4, male users are also identified to a larger degree than female users. However, this distribution has the largest percentage of unidentified user of all investigated datasets. The large number of unrecognized genders in all datasets may indicates that the method used for deriving gender should be improved. This will be further discussed in Chapter 6.

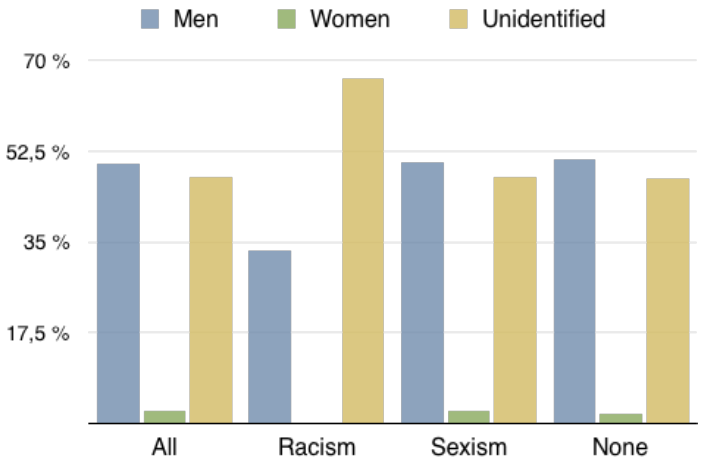


Figure 4.1: Gender distribution derived by Waseem and Hovy (2016)

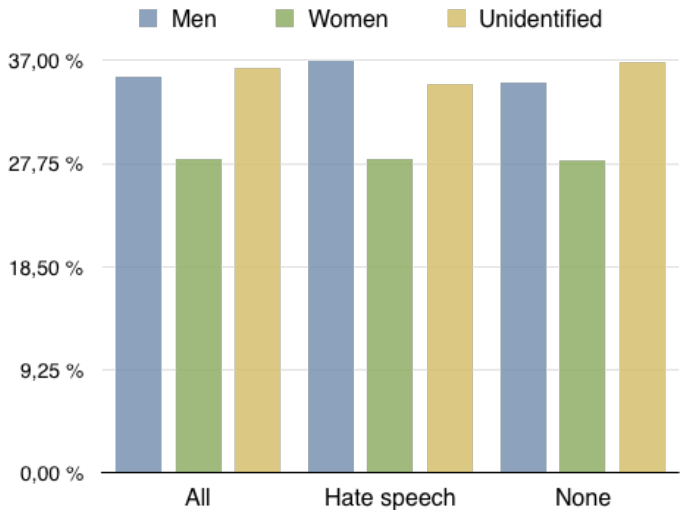


Figure 4.2: Gender distribution of users in dataset by Waseem and Hovy (2016)

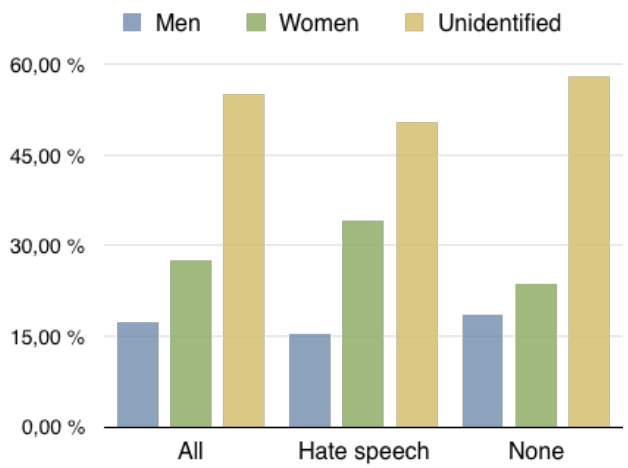


Figure 4.3: Gender distribution of users in dataset by Fortuna (2017)

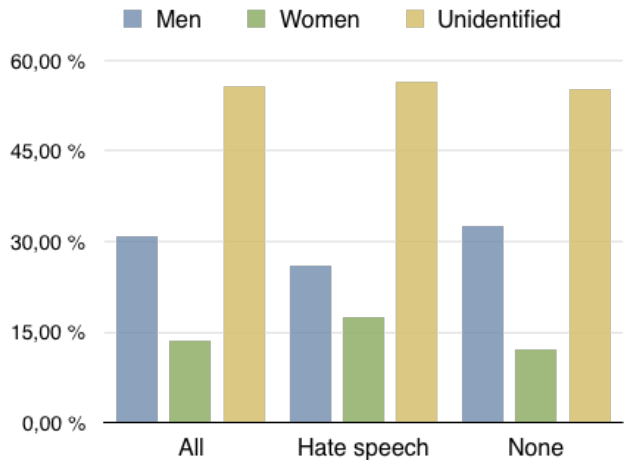


Figure 4.4: Gender distribution of users in dataset by Ross et al. (2016)

4.2.2 User Network

A users' network is here defined as the social network of a user on Twitter, i.e. who a user is following and who is following a user. Following and being followed by other users is a fundamental functionality on Twitter. Following refers to a user subscribing to another user's tweets, and similarly a user can have its own followers that subscribe to the user's own tweets. Chatzakou et al. (2017) found that network-based features were very useful in classifying aggressive user behavior. In their study, the authors investigated features such as the ratio of followers to friends, the extent to which users reciprocate the follower connection they receive from other users, and the user's tendency to cluster with others. Badjatiya et al. (2017) also plan to explore the importance of user network features in hate speech detection. This suggests that a user's network may be of significance in attempts to detect hate speech. Tweepy supports the retrieval of the fields *user.followers_count* and *user.friends_count*, that correspond to the number of people that a specific user is following and the number of users that are following a specific user, respectively.

In Figure 4.5, the relationship between a user's followers and friends in the dataset by Waseem and Hovy (2016) is illustrated. The majority of users form a cluster in the area below 10,000 following and 50,000 friends. Beyond this cluster, it appears as users of the "None" class are most common, with the exception of one outlier of the "Hate speech" class with about 228,00 followers and no friends. It is difficult to say whether this trend can be generalized, or is caused by the uneven number of users in the two target classes. Figure 4.6 shows the distribution of friends and followers for the users in the dataset by Fortuna (2017). A general observation is that the users of this dataset often tend to have more followers than friends. Furthermore, there is little that distinguishes the users of the two classes regarding the number of friends and followers. The number of users in the dataset by Ross et al. (2016) is considerably lower than the other datasets, and may explain the lower number of friends and followers for the users, as shown in Figure 4.7. There is an outlier in the "Hate speech" class with about 13,000 followers and 14,000 friends, but the rest of the users are somewhat evenly distributed.

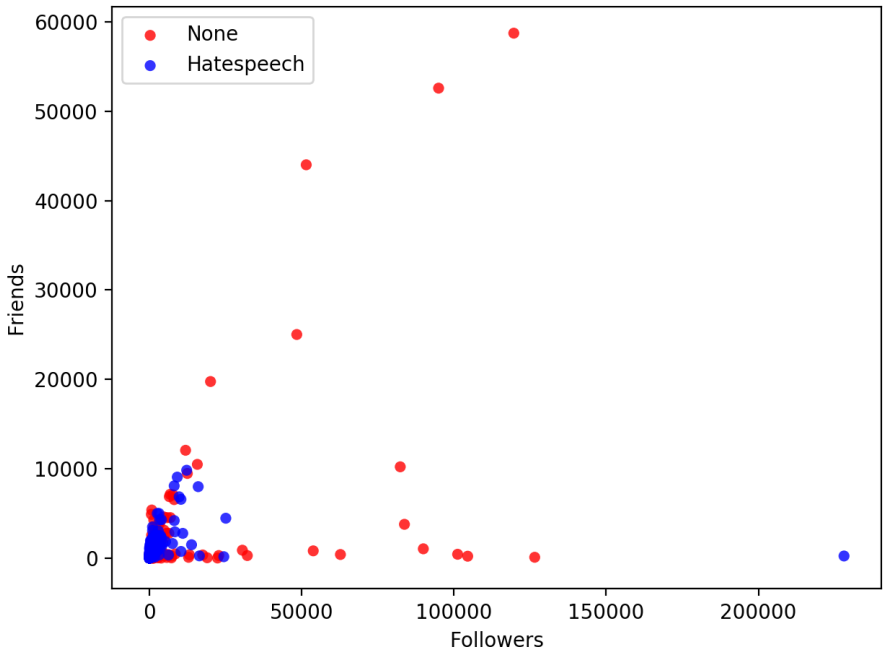


Figure 4.5: Distribution of users based on their network in dataset by Waseem and Hovy (2016)

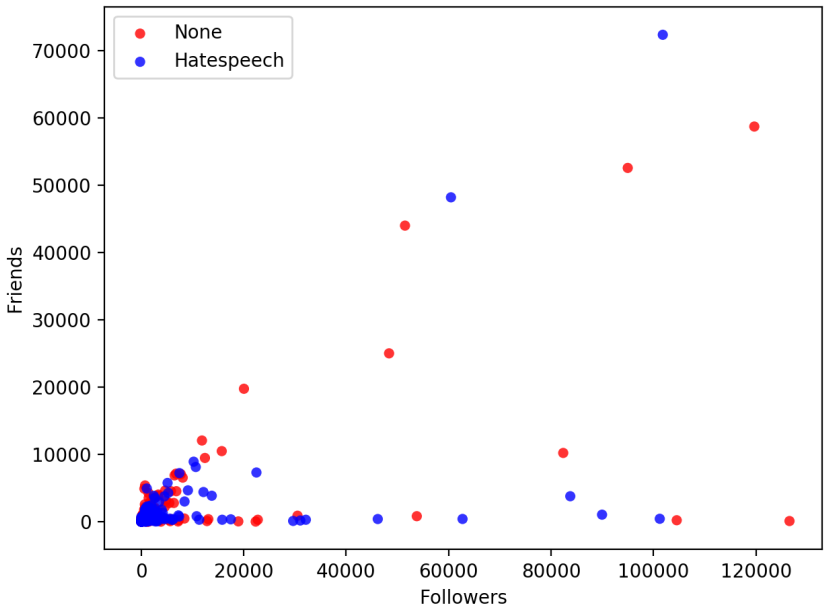


Figure 4.6: Distribution of users based on their network in dataset by Fortuna (2017)

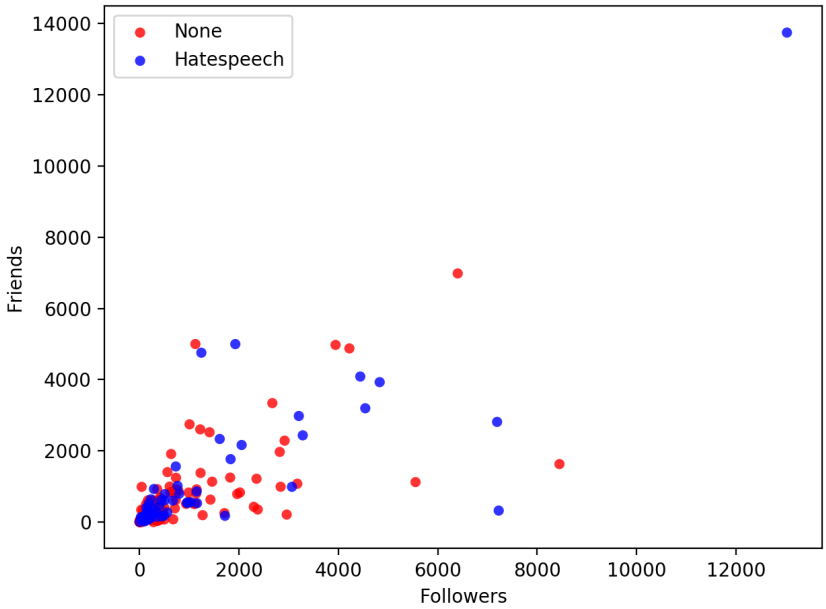


Figure 4.7: Distribution of users based on their network in dataset by Ross et al. (2016)

4.2.3 Activity

Table 3.2 contains several instances supporting a user's activity level as a feature. However, the related research suggests that both a high and low activity level can be related to posting hate speech content. Buckels et al. (2014) found that commenting frequency was positively associated with trolling enjoyment, and Cheng et al. (2015) suggested that frequently active users are often associated with anti-social behavior online. On the other hand, Wulczyn et al. (2017) found that users of both high and low activity levels are causing personal attacks on Wikipedia. Although these findings derive from studies that address different aspects of hate speech, it is interesting to examine whether the findings correspond to the characteristics of the users in the datasets used in this thesis. How a user's activity level is measured can differ, and examples of representations can be the number of posts by a user or the amount of time a user spends online. In this case the activity is defined by the available information that can be extracted through the Twitter API. Tweepy enables the retrieval of the number of all tweets a user has posted, and the number of "favorites" a user has given to tweets by other users, which corresponds to "likes" used in other online services. Therefore, a user's activity will in this case be determined by the fields *user.statuses_count* and *user.favourites_count*.

In Figure 4.8 the relationship between a user's total number of statuses and number of favourites given is illustrated. With the exception of one outlier in the "Hate speech" class with over 400,000 favourites and over 600,000 statuses, the majority of users in both classes of the dataset by Waseem and Hovy (2016) form a cluster below 50,000 favourites and 200,000 statuses. There is also a general tendency to have a larger number of statuses than favorites. In the dataset by Fortuna (2017), the users have also have a larger number of own statuses than likes given to others, as shown in Figure 4.9. The users of both target classes are somewhat evenly distributed, and in general the users of this dataset have posted below 200,000 tweets and given below 25,000 favourites. Similarly to the users in the two previously described datasets, the users in the dataset by Ross et al. (2016), in Figure 4.10, also tend to have more statuses than favourites. However, the total number of

statuses and favourites for the users are much lower in this dataset. Similarly to the findings investigating the users' network, there is no clear distinction between the activity characteristics of the users in the target classes.

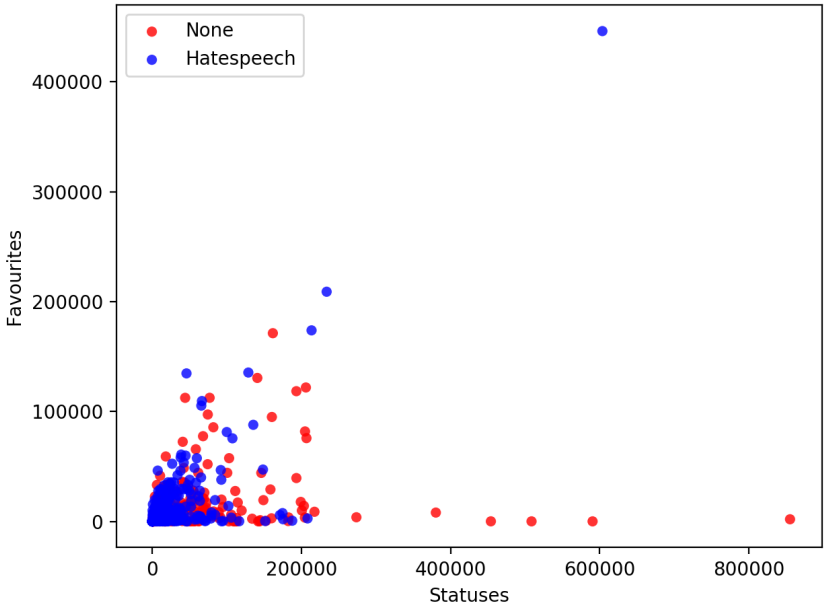


Figure 4.8: Distribution of users based on their activity in dataset by Waseem and Hovy (2016)

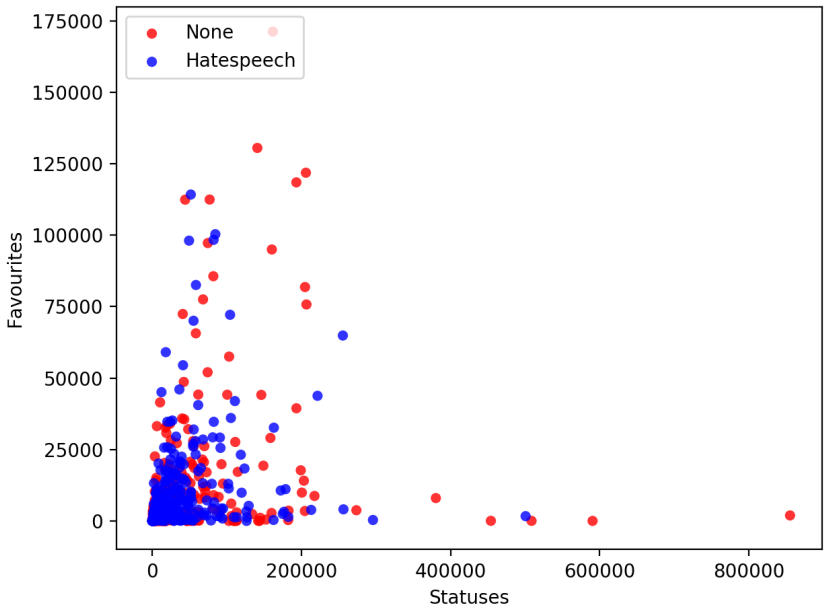


Figure 4.9: Distribution of users based on their activity in dataset by Fortuna (2017)

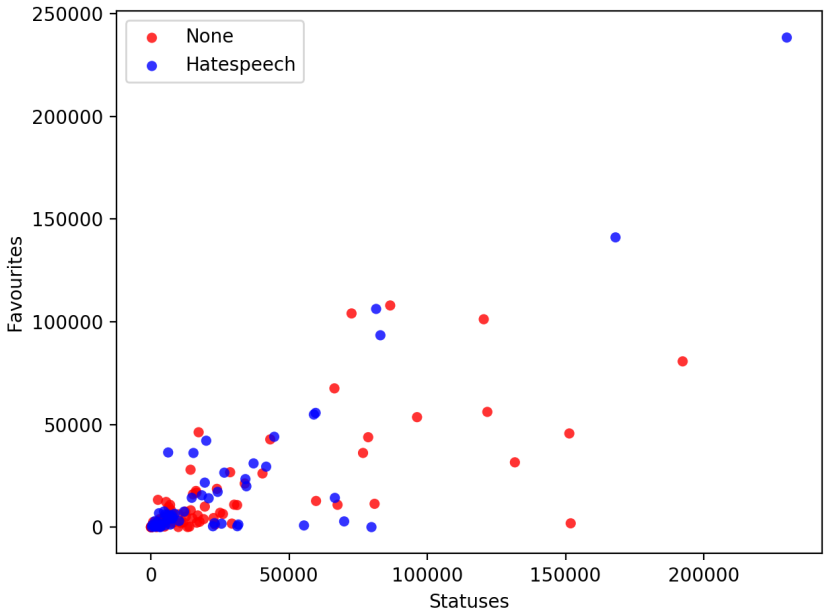


Figure 4.10: Distribution of users based on their activity in dataset by Ross et al. (2016)

4.2.4 User Profile

Twitter enables users to customize their own profile page, exemplified by changing the theme color, adding a profile picture or header picture. In addition, users can add a bio description, a geographical location or a link to a web page. Wulczyn et al. (2017) found that the personal attack prevalence among comments by anonymous users is higher than of registered users. Therefore, the elements of a user's profile that can be personalized were examined with the underlying assumption that personalizing the user profile elements is contradicting to remaining anonymity. The elements retrieved were the number of public lists a user has joined, geotagging of tweets, the profile image and whether or not the user has altered a default theme or background of the profile. These elements will in turn be described.

Lists are organized groups of Twitter accounts that can either be private or public. When made public, any Twitter user can subscribe to the list. *Listed_count* is the number of public lists that this user is a member of. While lists are included as a user profile characteristic, it would also be suitable as a part of the "Network" or "Activity" characteristics. The public list information was not included in the data analysis, but the information was still used in the classification experiments. When enabled, geotagging allows geographic data to be attached to a user's tweets. In Tweepy, this field is retrieved by using *geo_enabled*. A user has a default profile when the theme color or background of their user profile has not been altered. The background color is by default set to a blue color, but Twitter provides other alternative theme colors and the opportunity to enter an own hexadecimal color value. Tweepy allows retrieving of the field *default_profile*, which is true when no changes are made by the user. When true, *default_profile_image* indicates that a user has not uploaded an own profile image, and instead uses the default image provided by Twitter.

The users in the dataset by Waseem and Hovy (2016) are somewhat equally divided in the enabling and disabling of geotagging for both target classes, as seen in Table 4.7. The distribution is similar for the geotagging characteristic of users in

the dataset by Fortuna (2017) as well, shown in Table 4.8. However, the users in the dataset by Ross et al. (2016) have a noticeable difference regarding the distribution. As seen in Table 4.9, the majority of users in both target classes have disabled the geotagging of tweets, with a slightly higher percentage for the users in the "Hate speech" class. Nearly all the users in the three datasets have changed their profile image, suggesting that this is common for Twitter users in general. For all the datasets, the percentage of changed profile images is also marginally higher for the users in the "Hate speech" class than the users in the "None" class. Figure 4.7 illustrates a tendency of the users in the dataset by Waseem and Hovy (2016) to rather have a customized profile page than a standard, which corresponds with the profile characteristic of the users in the dataset by Fortuna (2017), in Table 4.8. The users in the dataset by Ross et al. (2016) are more equally distributed regarding the profile characteristic, as shown in Table 4.9

Feature	Values	"Hate speech" class	"None" class
Geotagging	Enabled	51.7%	48.6%
	Disabled	48.3%	51.4%
Profile	Default	39.9%	27.6%
	Changed	60.1%	72.4%
Profile image	Default	1.9%	3.8%
	Changed	98.1%	96.2%

Table 4.7: User profile characteristics of dataset by Waseem and Hovy (2016)

Feature	Values	"Hate speech" class	"None" class
Geo	Enabled	58.8%	58.2%
	Disabled	41.2%	41.8%
Profile	Default	24.9%	32.6%
	Changed	75.1%	67.4%
Profile image	Default	0.4%	1.8%
	Changed	99.6%	98.2%

Table 4.8: User profile characteristics of dataset by Fortuna (2017)

Feature	Values	“Hate speech”class	“None”class
Geo	Enabled	16.1%	26.6%
	Disabled	83.9%	73.4%
Profile	Default	50.0%	54.1%
	Changed	50.0%	45.9%
Profile image	Default	1.6%	1.8%
	Changed	98.4%	98.2%

Table 4.9: User profile characteristics of dataset by Ross et al. (2016)

5 Experiments and Results

As stated by Oates (2005), an experiment is a research strategy that serves to investigate a link between a factor and observed outcome. This chapter presents the experiments conducted for the purpose of investigating the possible effects user features have on the performance of hate speech classification. The first section describes the experimental plan and design. The second section presents the architecture of the classifier implemented for conducting experiments. Lastly, the performance of the classifier on the different datasets is presented, first with only simple surface features and then with user features. The implemented hate speech classifier was trained and tested using data from three different datasets, as presented in Chapter 4. As a result, some parts of this chapter are divided accordingly, to present the findings from each dataset in a structured fashion.

5.1 Experimental Plan

An experiment is designed to prove or disprove a hypothesis. The hypothesis of the experiments in this thesis was that user features had an impact on the performance of hate speech classification. The first part of the experiments concerned implementing a baseline hate speech classifier that was only based on the textual tweets from the datasets. This classifier served as a basis for comparison of results. Finding suitable preprocessing techniques and feature representations was included in the first part of the experiments, which will be presented in the next section. The performance of the baseline classifier was evaluated by training and testing on the datasets presented in Chapter 4.

The analysis of the datasets presented in the previous chapter indicated that none of the investigated user characteristics could be used to differentiate textual tweets annotated “Hate speech” and “None”. However, the impact of user features in detection may become more visible when tested through a classifier. The second part of the experiments concerned incorporating the features found in Chapter 4 to the baseline classifier. The performance of the model with user features was compared to the performance of the model without user features. Along with observing the overall effects of user feature inclusion, the impact of the individual features and feature subsets was also investigated. Too many irrelevant features can negatively affect the performance of the model, and it is therefore important to find the features that best represent the underlying problem and are correlated to the predicted instances. As an attempt to gain more insight into the effect of user feature inclusion, the distribution of misclassified instances was analyzed.

5.2 Model Architecture

In order to measure the effects user features have on hate speech classification, a simple text classifier was implemented. This section describes the architecture of the hate speech classifier. Figure 5.1 illustrates the general components of the architecture, from the tweets in the datasets to the classifier predictions. This section will describe each of the architectural components in turn, from preprocessing to classification.

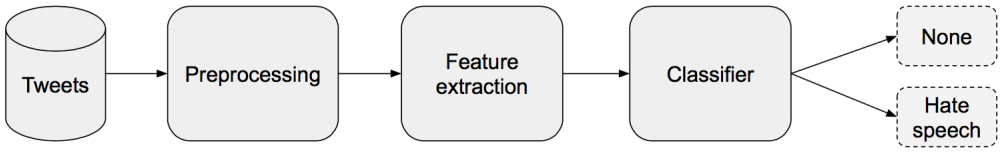


Figure 5.1: Outline of the hate speech classifier architecture

5.2.1 Preprocessing

Text processing is a difficult task due to the noise contained in natural language. In addition to the challenges of understanding natural language, Twitter also introduces domain-specific challenges for text processing. Firstly, the limit of 280 characters in a tweet increases the use of abbreviations. Second, including non-textual content such as URLs, images, user mentions and retweets is common, causing the need for a larger amount of interpretation and structuring. Text preprocessing should be done with care, to avoid losing any important features. Therefore, a simple preprocessing phase was preferred. The providers of the datasets had already conducted a filtering when collecting the data, as discussed in Section 4.1, but the tweets themselves had not been processed. The NLTK library, described in Section 2.4 was used for preprocessing of the data. The preprocessing steps consisted of:

- **Remove Twitter specific information** - Twitter specific information such as user mentions, retweets and URLs were removed from the tweets. This type of text may provide information about the context, but this is considered to be outside the scope of this thesis. The actual hashtag symbol was removed from hashtags, transforming the content of hashtags to simple words. Emoticons were also removed to only retain textual content.

- **Tokenization** - All tweets in the datasets were tokenized and stripped of punctuation and special characters.

- **Lowercasing** - In the process of tokenizations all words were converted to lowercase, to make it easier to compare words and to reduce the sample size.

- **Stop word removal** - Stop words were removed as they contribute little to the meaning of the tweet and to differentiate classes. Different stop word lists were used for the datasets, due the the different languages. The list of stop words removed can be found in Appendix 7.2.

As discussed in Section 2.3.1, stemming and lemmatization are also commonly

used preprocessing techniques. However, these techniques have been excluded due to the uncertain effects related to these methods.

5.2.2 Feature Extraction and Representation

Before experimenting with the user features, a simple text classifier was created by using the tweets to make predictions. Therefore, extracting features from the tweets and finding meaningful representations was necessary. As mentioned in Section 3.2.3, n-grams have been shown to be very useful in text classification, and therefore both character n-grams and word n-grams were tested to represent the textual content of the tweets. N-gram ranges up to n=6 were tested, but higher values of n were not considered due to the computational effort required. The most suitable type of n-gram and n-gram range were explored through a grid search, and it was found that different alternatives for representing the tweets suited the different data. A grid search exhaustively considers all combinations of parameters of a given set to find the parameters that contribute to the best model performance. The grid search uses cross-validation on a part of the dataset to evaluate and select the best settings. The exact features used for each dataset will be presented in the next section along with the classification results.

A tf-idf approach was used to represent the n-gram features. As explained in Section 2.3.1, tf-idf is a representation of features where weights are assigned to terms according to their importance to a document in a corpus. As a result, this representation highlights words that are distinct for a given document.

5.2.3 Classification Model

A logistic regression model was chosen for classification due to its simplicity and its common usage in NLP classification tasks. As the task was not to implement the best performing classifier but to test the effects of user features, no other classification models were tested. The classification task was binary, where the model

attempted to classify a tweet in the class “Hate speech” or “None”. The dataset was initially split into training data and test data, as shown in Figure 5.2, to ensure that the model performance was evaluated on unseen data. Selecting the best features and hyperparameters of a classification model can have significant effects on the model performance. There exists several methods for finding the best parameters, from a manual search to randomly testing all possible combinations of a given parameter set. A grid search with a 10-fold cross-validation technique was chosen for selecting the model parameters. This technique divides the data into 10 equal sized sections, and then the model is trained 10 times where a different section of the data is left out and tested on each time. The performance results for each of the 10 iterations are then averaged to represent the final predictive model. Only the training data was used in cross-validation when searching for the best hyperparameters. Selecting hyperparameters based on the whole dataset introduces a bias because the algorithm has already seen the test set, and the accuracy estimate is likely inflated (Refaeilzadeh et al., 2007). Therefore, hyperparameter selection was performed inside the cross-validation loop for the classifier. Finally, the classification model with the chosen hyperparameters was evaluated on the unseen test set.

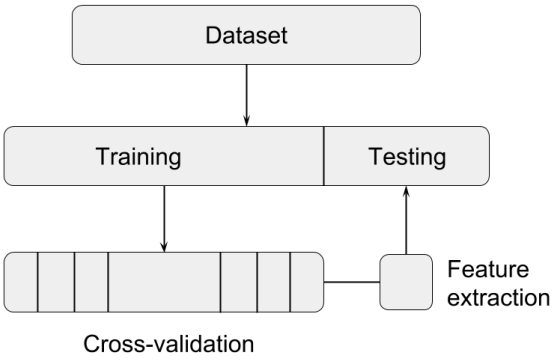


Figure 5.2: Representation of training set and test set splitting

5.3 Results

The classification model was trained and tested on the datasets presented in Section 4.1. The tables in this section present the performance of the classifier in terms of precision, recall and macro F_1-score. The section first presents the results from the classification model with only n-gram features, and then the results from incorporating user features as well.

5.3.1 Classifier with Text Features

The first part of the experiment consisted of implementing a hate speech classifier that only used textual features from the tweets in the datasets. This section presents the results of the grid search and the baseline classifier performance.

Waseem and Hovy (2016)

The dataset provided by Waseem and Hovy (2016) contained 15,727 available tweets, which were split into a training set of 11,008 tweets and a test set containing 4,719 tweets. A grid search over a subset of parameters for n-gram features found that character n-grams in range [1,5] provided the best performance, as shown in Table 5.1. Table 5.2 shows the performance metrics of the model on this dataset, where 0.82 was the macro average F_1-score for both classes. Both the precision value and the recall value are higher for the "None" class, with recall having the largest difference. However, the recall value for the "Hate speech" class obtained for this dataset is higher than for any other datasets, most probably due to the larger amount of available data, and therefore also a greater amount of model training.

n-gram range	[1,1]	[1,2]	[1,3]	[1,4]	[1,5]	[1,6]
Word	0.81659	0.81677	0.81468	0.81186	0.81168	0.81096
Char	0.73992	0.80205	0.82013	0.82258	**0.82476**	0.82376

Table 5.1: Grid search of n-gram parameters with dataset by Waseem and Hovy (2016). The scores are the average of all cross-validation folds for a single combination of parameters

Class	Precision	Recall	F1-score	Support
None	0.83	0.94	0.89	1,444
Hate speech	0.82	0.58	0.68	3,275
Average	0.83	0.83	0.82	4,719

Table 5.2: Baseline model performance on test data with dataset from Waseem and Hovy (2016)

Fortuna (2017)

A total of 3,059 tweets from the dataset developed by Fortuna (2017) were used to train and test the classification model, where the training set contained 2,636 tweets and the test set contained 423 tweets. Figure 5.3 shows that word unigrams yielded the best performance for this dataset. Trained with a 10-fold cross validation and the specified n-gram parameters, the macro average F_1-score obtained for the test data for both target classes was 0.77. Further, as shown in Table 5.4, the precision value obtained for the "Hate speech" class is slightly higher than for the "None" class. However, the recall value for the "Hate speech" class is much lower than the "None" class, which highly impacts the F_1-score.

n-gram range	[1,1]	[1,2]	[1,3]	[1,4]	[1,5]	[1,6]
Word	**0.77688**	0.77181	0.76876	0.76673	0.76369	0.76116
Char	0.69270	0.73834	0.75254	0.76572	0.76978	0.77586

Table 5.3: Grid search of n-gram parameters with dataset by Fortuna (2017). The scores are the average of all cross-validation folds for a single combination of parameters

Class	Precision	Recall	F1-score	Support
None	0.79	0.96	0.87	297
Hate speech	0.82	0.40	0.53	126
Average	0.80	0.79	0.77	423

Table 5.4: Baseline model performance on test data with dataset from Fortuna (2017)

Ross et al. (2016)

The dataset provided by Ross et al. (2016) is considerably smaller than the other datasets, containing only 341 tweets. As shown in Table 5.6, the classifier was only able to identify 0.03% of the instances of the "Hate speech" class in the test data. This is most likely due to the few instances in the dataset, resulting in an insufficient amount of training. The dataset was split into a training set containing 238 tweets and a test set containing 103 tweets. A grid search of the n-gram parameters showed a character n-gram with the range [1,2] produced the best results, as shown in Table 5.5. With a 10-fold cross validation and word unigrams, the model received a macro average F_1-score of 0.57. It may be worth mentioning that this dataset was not initially developed for classification in the study by Ross et al. (2016), but for investigating the annotation reliability of hate speech. The study concluded that the presence of hate speech perhaps should not be considered a binary yer-or-no decision; however, this is how the current classification model is operating.

n-gram range	[1,1]	[1,2]	[1,3]	[1,4]	[1,5]	[1,6]
Word	0.72269	0.71849	0.72269	0.72269	0.72269	0.72269
Char	0.71849	**0.72689**	0.71008	0.71008	0.71429	0.71429

Table 5.5: Grid search of n-gram parameters with dataset by Ross et al. (2016). The scores are the average of all cross-validation folds for a single combination of parameters

Class	Precision	Recall	F1-score	Support
None	0.68	0.99	0.81	70
Hate speech	0.50	0.03	0.06	33
Average	0.62	0.68	0.57	103

Table 5.6: Baseline model performance on test data with dataset from Ross et al. (2016)

5.3.2 Classifier with Text Features and User Features

In the second part of the experiments, the classifier was expanded to incorporate various user features and subsets in addition to n-grams. Table 5.7 presents the user features experimented with, grouped by the sections presented in Chapter 4.

	Feature	Type	Values
Gender	Male	Boolean	[0,1]
	Female	Boolean	[0,1]
Network	Followers	Integer	$\mathbb{N}$
	Friends	Integer	$\mathbb{N}$
Activity	Statuses	Integer	$\mathbb{N}$
	Favourites	Integer	$\mathbb{N}$
Profile	Geo Enabled	Boolean	[0,1]
	Public Lists	Integer	$\mathbb{N}$
	Default Profile	Boolean	[0,1]
	Default Image	Boolean	[0,1]

Table 5.7: Overview of user features subsets

Waseem and Hovy (2016)

Table 5.8 shows the average performance of the model with only n-gram features and then n-grams along with various subsets of user features. The inclusion of all user features yielded in the largest improvement from the baseline classifier using simple surface features. Of the subsets tested, it was the inclusion of "Network"

that produced the largest improvement for precision, recall and F_1-score to the baseline classifier. "Gender" did not improve performance at all, and the "Activity" and "Profile" feature subsets provided very slight improvements. In investigation of the individual features, each of the features were tested individually along with the n-gram features, and the findings are shown in Figure 5.3. More than half of the features did not impact performance at all when included. With "Default profile" and "Geo enabled" the average F_1-score increased by 0.1, while "Female", "Followers" and "Public lists" had the most impact and increased F_1-score by 0.2.

	Precision	Recall	F1-score
n-grams	0.83	0.83	0.82
+ all features	0.86	0.86	0.86
+ gender	0.83	0.83	0.82
+ network	0.84	0.85	0.84
+ activity	0.83	0.84	0.83
+ profile	0.83	0.83	0.83

Table 5.8: Evaluation scores of the classifier with n-grams and different user feature sets on the dataset by Waseem and Hovy (2016)

Fortuna (2017)

The incorporation of all user features in the classification on the dataset by Fortuna (2017) resulted in a slightly worsened classifier performance, as shown in Table 5.9. This was also the case for inclusion of the "Activity" subset. On the other hand, the inclusion of "Network" improved the performance for all evaluation metrics. Including the "Gender" and "Profile" subsets received the same F_1-scores as the baseline classifier. Of the individual features, "Followers" and "Geo enabled" resulted in the largest increase of the F_1-score when used in combination with n-gram features, as shown in Figure 5.4. In addition, the inclusion of "Public lists" also slightly improved the F_1-score. It is interesting to notice that the inclusion of "Female" and "Statuses" actually result in a worsened model performance. This may be caused by several factors, and will be discussed in Chapter 6.

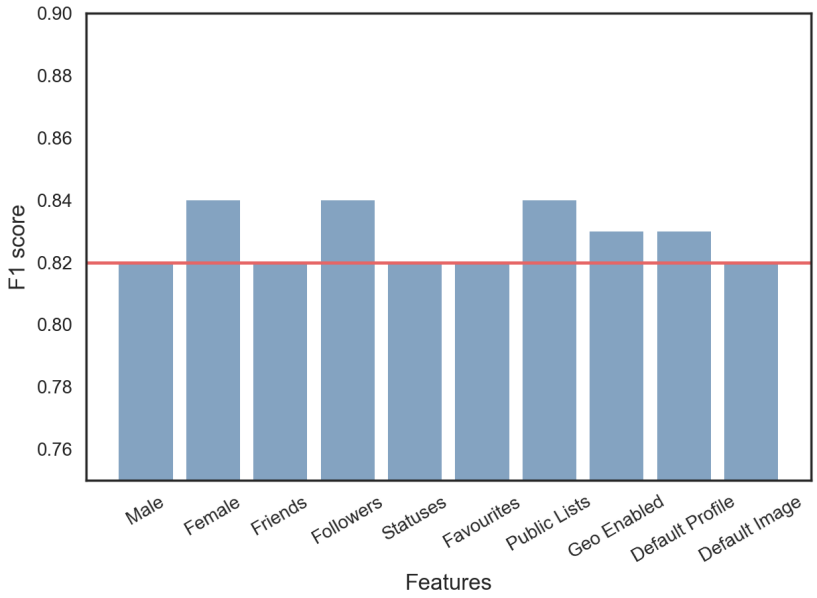


Figure 5.3: F_1-scores of individual features along with n-grams on dataset by Waseem and Hovy (2016). The red line represents the average F_1-score of only including n-grams.

	Precision	Recall	F1-score
n-grams	0.80	0.79	0.77
+ all features	0.79	0.79	0.76
+ gender	0.80	0.79	0.77
+ network	0.81	0.81	0.79
+ activity	0.79	0.79	0.76
+ profile	0.80	0.80	0.77

Table 5.9: Evaluation scores of the classifier with n-grams and different user feature sets on the dataset by Fortuna (2017)

Ross (2016)

By only using word unigrams, the classifier only received a recall value of 0.03 for the hate speech class of the dataset by Ross et al. (2016). As shown in Table 5.10, all tested feature subsets resulted in improvement of the precision value and on

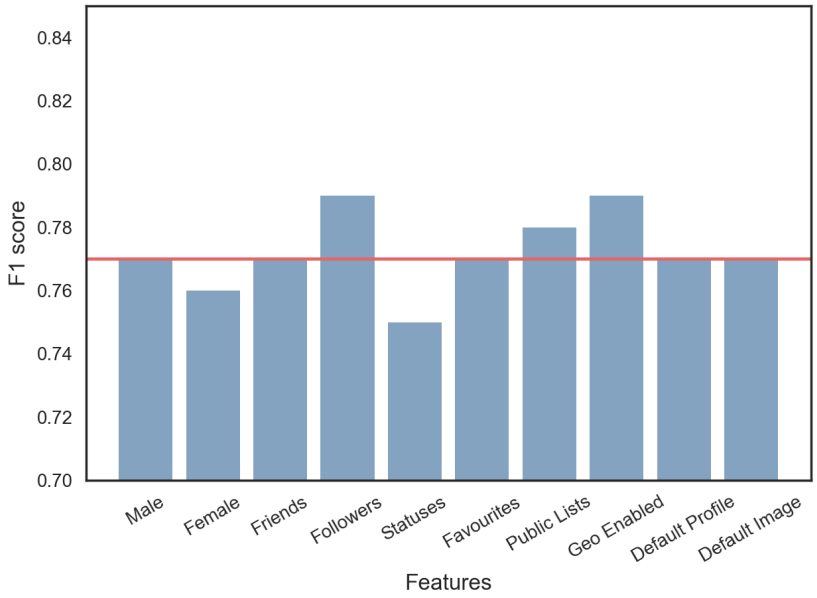


Figure 5.4: F_1-scores of individual features along with n-grams on dataset by Fortuna (2017). The red line represents the average F_1-score of only including n-grams.

the average F_1-score. The inclusion of all the features and the subsets "Network" and "Activity" increased the average F_1-score by 0.02. "Gender" increased the F_1-score by 0.05, and "Profile" resulted in the largest impact by increasing the F_1-score by 0.07. These results are consistent with the testing of the individual features shown in Figure 5.5, where "Male", "Female" and "Profile" have the largest impact on performance. Of the individual features included, only "Default image" and "Default profile" did not lead to any improvement of the basline F_1-score.

	Precision	Recall	F1-score
n-grams	0.62	0.68	0.57
+ all features	0.63	0.68	0.59
+ gender	0.69	0.70	0.62
+ network	0.63	0.68	0.59
+ activity	0.68	0.69	0.59
+ profile	0.71	0.71	0.64

Table 5.10: Evaluation scores of the classifier with n-grams and different user feature sets on the dataset by Ross et al. (2016)

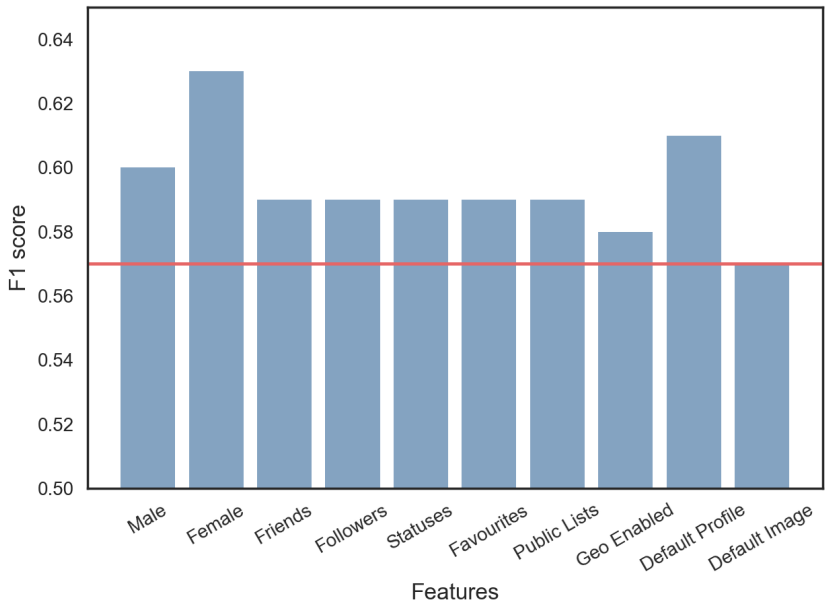


Figure 5.5: F_1-scores of individual features along with n-grams on dataset by Ross et al. (2016). The red line represents the average F_1-score of only including n-grams.

6 Evaluation and Discussion

The first section of this chapter evaluates the experimental results presented in Chapter 5, and explains how they may have been impacted. Then, the findings are discussed with regards to the research questions presented. Limitations and merits of the work conducted will also be presented throughout this chapter.

6.1 Evaluation

This section provides an interpretation and evaluation of the experimental results. The section will cover the different aspects of the experiments, and will be divided accordingly.

Datasets

Several datasets have been used both in the analysis of user characteristics and for training and testing the classification model. While a comparison of the results obtained for each dataset may result in a discovery of similarities and trends, it is most likely also a rationale for differences in classifier results. The amount and type of data is an essential aspect for machine learning models, and heavily impacts a models performance. One limitation of using several datasets was that each dataset has different annotations by different annotators, and that they were developed for different subtasks. This makes a blind comparison of results unsafe, as there might be differences in the interpretations of hate speech. The possible differences

of interpretations may also be affected by the different languages and geographical areas of the users in the datasets. The main difference of the datasets is the size. This affects the amount of model training, which is probably the main impact of the different results. The model's performance on the dataset by Ross et al. (2016) was worse than on the datasets by Fortuna (2017), which was again worse than on the dataset by Waseem and Hovy (2016). The unbalanced distribution of the target classes is however present for all datasets, and causes the similarity of poor performance metrics for the "Hate speech" class for all datasets. All though the datasets cause differences of model performance, it also allows for a deeper insight into the topic.

In the analysis of the datasets, there was a trend that users who had posted tweets with hate speech annotations appeared similar to those who had not, based on the investigated characteristics. This similarity was present for all datasets, and might suggest that this is generally the case for Twitter users, even across different geographical areas.

User Information

As the goal of this thesis was to investigate the possible effects user features in hate speech classification, the Twitter users in the datasets have been of high importance. The available user information was bounded by the Twitter API, meaning that some information presented by related studies was not applicable in the experiments. A desire to retrieve and explore many different types of information meant that the information was not critically evaluated for use in hate speech classification. While some features investigated had basis in the findings from related studies, such as "Gender", "Network" and "Response", other features were based on an exploratory motive. This becomes clear in the results, as many features did not lead to any improved performance. One important aspect to consider is that the users who are in the "None" class may still have written tweets with hate speech content, that are not included in the datasets. Therefore a strict distinguishing of users in the "Hate speech" class and the "None" class should be

avoided. It should again be mentioned that the identity of the users in the datasets have not been exposed, and that anonymity was retained through the analysis and experiments.

Feature Representation and Selection

Only simple surface features, or n-grams, were chosen to represent the tweets in the classifier. All though other text-based features could have been included for a possible enhanced performance, the test results show that the n-grams operate relatively well on their own, at least where there was sufficient training data. An investigation of the highest scoring text features is presented in Table 6.1, with the 10 highest scoring words from the tweets in each dataset, based on univariate statistical tests[1]. In univariate feature selection, each feature is evaluated independently with respect to the target class. The words show that hate speech related words are often considered important, and shows the potential that text alone has for classification. The words presented are based on classification of both target classes, and not hate speech specifically.

Waseem and Hovy (2016)	Fortuna (2017)	Ross et al. (2016)
call	burra	asylanten
female	feia	bekommen
girls	gorda	frauen
islam	homem	hoax
mkr	mulher	hungerstreik
mohammed	mulherdeverdade	karneval
muslims	orgulho	kiel
notsexist	sapatao	menschenwürde
sexist	sapatão	rapefugees
women	ser	verstoß

Table 6.1: Highest scoring text features in all datasets

[1] http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html#sklearn.feature_selection.SelectKBest

By systematically investigating individual user features and subsets, it became clear how each may independently impact the results of the textual classifier. This is useful for a further investigation of those features or subsets that singularly included outperform the others. The downside is that possible important relations between the features was not captured. An example is the "Activity" subset used with the tweets from the dataset by Waseem and Hovy (2016), which combined with n-grams slightly improves the F1-score by 0.1. However, there is no impact of including "Statuses" and "Favourites" individually. This shows that combinations of features can be of importance, even when the features individually do not seem to contribute.

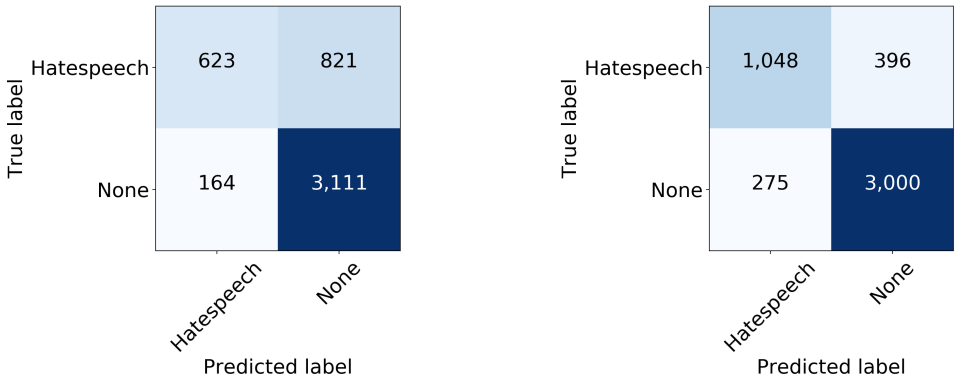
Classification Results

Regarding the implemented classification model, several alternatives were considered, from the text preprocessing procedure to model topology. However, only one solution was implemented, and alternatives were therefore not evaluated and compared. The solution was based on findings and experience from related work and project convenience. The preprocessing steps chosen were simple, but effectively reduced the amount of noise contained in the original tweets. Further preprocessing steps, such as stemming or lemmatization, were discarded due to the unknown effects. Logistic regression was also chosen as the model due to experience in related work, as presented in Chapter 3. With the intention of investigating the effects of user features in hate speech classification, the primary focus of the experiments was not to optimize the classifier. All though a grid search was used choosing the best parameters, other measures could have been made to improve the overall performance. The grid search of the best n-gram parameters resulted in different combinations for the classifier with only textual features, and when user features were included. This was the case for all the datasets by Waseem and Hovy (2016) and Ross et al. (2016), as shown in Table 6.2. This demonstrates that the grid search with cross-validation was useful for finding the most appropriate text representations in various settings.

Dataset	Text features		User features		
	Type	Range	Type	Range	Features
Waseem and Hovy (2016)	Char	[1,5]	Char	[1,6]	All
Fortuna (2017)	Word	[1,1]	Word	[1,1]	Network
Ross et al. (2016)	Char	[1,2]	Char	[1,1]	Profile

Table 6.2: Grid search variations for n-gram parameters with the baseline classifier and the classifier with user features resulting in the largest improvement for each dataset.

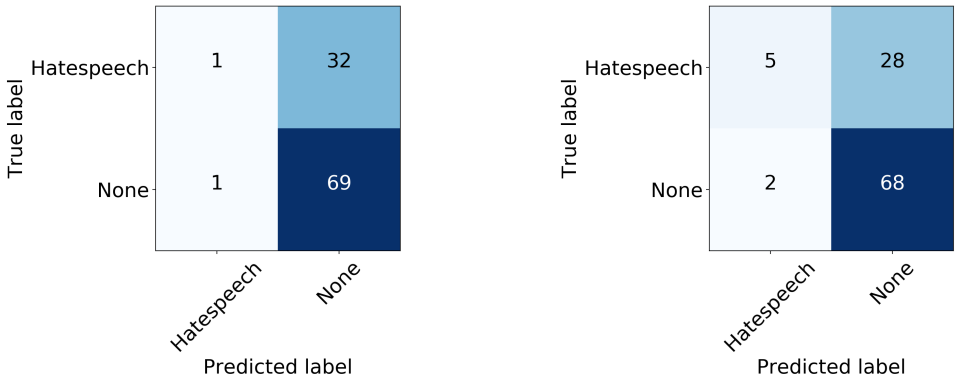
Some measures have been taken to avoid model overfitting, such as cross-validation and splitting the data into training and test sets. However, it may still be the case that the model is prone to overfitting. This has not been thoroughly investigated due to the time limits of the project. The results are also heavily affected by the uneven distribution of instances in the target classes. This is shown by the significantly lower F_1-scores for the "Hate speech" class than the "None" class for all datasets. The uneven distribution of instances is also reflected through the confusion matrices, illustrating the number of correctly and incorrectly classified instances. Figure 6.1 shows the confusion matrices for the classifier on the dataset by Waseem and Hovy (2016) with textual features and with all the user features, which yielded the best classifier performance on this dataset. A significant difference between the matrices is the increased number of correctly labeled "Hate speech" instances in Figure 6.1(b). Additionally, the matrices show that the introduction of all user features introduces a decreased number of correctly labeled instances of the "None" class. This pattern is also observed in Figure 6.2, showing the confusion matrices for the classifier tested on the dataset by Ross et al. (2016). However, the confusion matrices generated from the classifier on the dataset by Fortuna (2017) reflect other differences. Figure 6.3 show that there is actually a decrease of correctly labeled "Hate speech" instances when user features were introduced, and a slight increase of correctly labeled "None" instances. Thus, the reason for the increased F_1-score when using both n-gram and network-based features in the classification model was due to the increased number of correctly labeled instances in the "None" class, thereby increasing the recall value of the class and the overall F_1-score.



(a) Classifier with only n-gram features

(b) Classifier with n-gram features and all user features

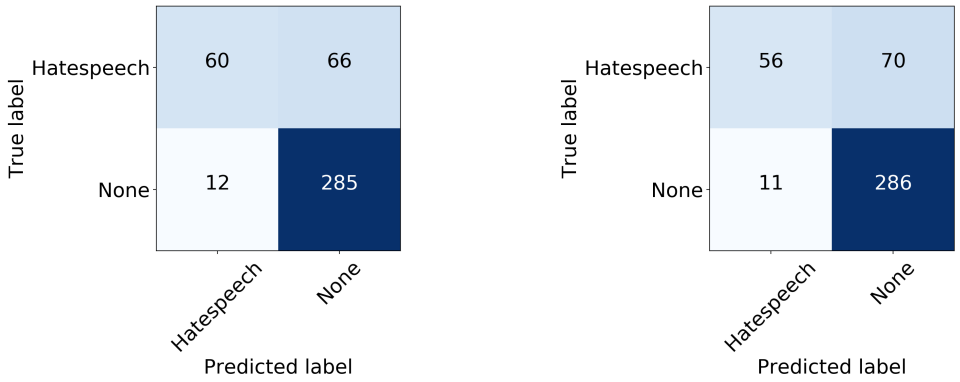
Figure 6.1: Confusion matrices for the classifier trained and tested on the dataset by Waseem and Hovy (2016)



(a) Classifier with only n-gram features

(b) Classifier with n-gram features and the profile subset

Figure 6.2: Confusion matrices for the classifier trained and tested on the dataset by Ross et al. (2016)



(a) Classifier with only n-gram features

(b) Classifier with n-gram features and the network subset

Figure 6.3: Confusion matrices for the classifier trained and tested on the dataset by Fortuna (2017)

6.2 Discussion

The goal of this thesis was to investigate the impact of user features in automatic hate speech classification. Two research questions were formulated for reaching the goal, which will be addressed in this section.

Research Question 1 *What does the literature and data suggest as promising user information for hate speech detection?*

With regards to Research Question 1, existing research presents characteristics about the users who write and act hatefully online, and results from studies that have investigated actual effects of user features in classification. These findings are summarized and presented in Table 3.2. Several longitudinal studies have investigated typical characteristic or personal traits of users. These types of studies allow observations of the nature and trends of users in online communities, and often investigate specific types of users, such as frequently banned users as in Cheng et al. (2015). Although these studies provide good and meaningful indicators of who the hateful users are, it can be difficult to translate such information into features for automatic detection and the information may not be easily observable or reflected through an online profile.

From the analysis conducted, no correlations were found with tweets annotated as hate speech and a set of user characteristics including, gender, network, activity and a user's Twitter profile. In similar matters, Cheng et al. (2017) suggested that ordinary people can engage in trolling behavior under specific circumstances, and that many undesirable posts are actually written by ordinary users. More specifically, the study found that exposure to prior troll posts was an important trigger mechanism for trolling. Similarly, Wulczyn et al. (2017) proposed that personal attacks cluster in time, caused by attacks triggering other attacks. Therefore, a proposition for a more effective hate speech detection would be to examine how users are affected in the online communities, in addition to who they are and what they have written. This is also in line with a framework Twitter has introduced[2],

[2]https://investor.twitterinc.com

which is a holistic approach to detect spam and abuse on Twitter by viewing such challenges through the larger lens of the total health of conversations.

Other studies propose specific information or types of features that are useful or seem promising for hate speech detection. These are important findings and indications that other researchers can explore in further research. Waseem and Hovy (2016) found that character n-grams in conjunction with gender brought a slight model improvement. However, the results from the classifier experiments on the same dataset in this thesis did not support these claims. Chatzakou et al. (2017) proposed that network-based features were very indicative for hate speech classification. These findings are reflected through the outcome of the experiments in this thesis, where the "Network" feature subset caused slight model improvement on all datasets.

Research Question 2 *What are the effects of incorporating user features in hate speech classification?*

In relation to Research Question 2, the experimental results indicate that user features can be used in addition to textual features to slightly increase the predictive power and improve the performance of the baseline model. While the classifier using only n-gram features performed relatively well, as also supported by several other researchers, the inclusion of some user features still results in a slightly improved model performance. The most prominent user features and feature subsets were different for each of the datasets, and it is therefore difficult to generalize the findings. This may be due to the fact that the datasets were created for various research tasks, and that different features are appropriate for different subtasks. It is therefore important that more studies aim to map and compare the effects of user features across various datasets, to find the most appropriate features for different settings. However, the "Network" feature subset was found to be useful and improved the F_1-score for all datasets when used together with n-gram features. This corresponds to the findings of Chatzakou et al. (2017), where network-based features are very effective in classifying aggressive user behavior. Although "Network" consists of different types of information, one can argue that a user's social network on Twitter is important. The individual feature "Followers" consistently

improved the F_1-score on all datasets, which may be the reason for the usefulness of the "Network" subset. "Public Lists" and "Geotagging" were also individual features with improvements on all datasets. In the experiments, it was also experienced that some of the features were ineffective on their own, but worked well when used in combination with other features. This is supported by Guyon and Elisseeff (2003), who experience that variables that are useless by themselves can provide significant performance improvement when used with together or with other variables. The potential for incorporating user features seems promising, however there is still need for investigations of relations and comparisons of actual effects.

However, the incorporation of user features may at times also lead to a worsened model performance. This was experienced in the experiments on the dataset by Fortuna (2017), when including all user features and when including individual features or subsets. For individual features, this may be caused by the model interpreting the features as noise rather than a representation of the problem – thereby confusing the model. In their study, Waseem and Hovy (2016) also experienced that the addition of user features to text features was detrimental to the performance of the classifier, and possibly caused by the lack of coverage of the features tested. When the inclusion of all or several user features results in worsened performance, there is a possibility that the model is prone to the "curse of dimensionality". The curse of dimensionality is a concept introduced by Richard E. Bellman, which in machine learning occurs when the dimensionality of a problem becomes too complex for the model and thereby decreasing model performance. Introducing several features, or dimensions, to a classifier increases the data sparsity and can make it difficult to ascertain a pattern. Therefore, it is important that only the most indicative features are included in methods for hate speech detection, to avoid the curse of dimensionality.

As previously mentioned, Waseem and Hovy (2016) found that gender brought the most improvement of a set of tested features in hate speech classification. When experimenting with the classifier on the same dataset developed by Waseem and Hovy (2016), it was actually found that "Gender" was the only feature subset

that did not lead to any improved F_1-score. While other user features were easily retrievable through the Twitter API, the users' gender was derived from a comparative method, as explained in Section 4.2. The inconsistent impact of the gender subset and individual features on the different datasets may have been caused by this method, and the fact that the method is unable to identify the gender of a large amount of users in all datasets. A better approach, perhaps in combination with other gender identification methods, should be applied to properly investigate the impact gender has in hate speech detection. As of now, it can be argued that gender is not a useful feature to use where gender cannot be directly extracted.

7 Conclusion and Future Work

7.1 Conclusion

Hate speech is a growing problem for online communities allowing user generated content, and existing methods for handling such unintended behavior are still not efficient enough. Several studies aim to assist in developing effective tools, and the research field is growing. However, there are several challenges linked to the detection of harmful online behavior, such as detection beyond simply recognizing offensive words. In addition, there is a need for further investigation of the utilization and contribution of various features. Aiming to address this gap and to provide useful experience and insight, the work conducted in this thesis investigated the potential and effect of including user features in hate speech classification, focusing on the Twitter platform.

By reviewing existing literature related to users and hateful behavior online, an overview of information potentially applicable in detection methods was established. Existing literature presents a variety of characteristic traits and specific features that can be of importance, and this overview can be used by future researchers as a starting point for investigating how this information can be utilized and represented in detection methods. This thesis also contributed with a quantitative analysis of three different datasets based on Twitter. This analysis aimed to investigate various characteristics of users, based on their annotated tweets. The results indicated that there were no particular characteristics distinguishing the users who have had tweets annotated as hate speech and those who have seemingly not. In combination with findings from other research, a promising approach can

therefore be to not only consider who the users are or what they have written, but also how they are affected by surrounding factors in their online communities. Based on the findings from existing literature, the analysis of Twitter user characteristics, and the availability through the Twitter API, a set of user features were chosen for further investigation. Systematically incorporating the user features into a hate speech classifier in conjunction with n-gram features, allowed observations of the effects of individual features and feature subsets. The experiments were conducted by training and testing the classification model on three different datasets, allowing deeper insights and grounds for comparison of findings.

A logistic regression classifier using n-gram features provided a solid foundation for hate speech detection, given a sufficient amount of data. The experimental results also showed that the inclusion of specific user features, in addition to n-grams, caused a slight improvement of the baseline classifier performance. Of all tested feature subsets, only "Network" caused improvement of the classifier performance on all datasets, corresponding to the findings of Chatzakou et al. (2017) who found network-based features to be powerful in detecting aggressive behavior. This subset improvement may have been affected by the individual feature "Followers", which also increased the F_1-score on all datasets. The other subsets had inconsistent effects on the different datasets, suggesting that the impact is highly dependent on the data or the subtask the data was created for, and that more research is necessary before drawing conclusions. Gender was one of the features with inconsistent effects on the datasets. However, this feature was derived and not directly retrieved from the user profile, and the effects may have been caused by this. The experiments also found that the inclusion of some features was detrimental to model performance. Some of the individual features may have been considered noisy to the classifier, and the inclusion of several features possibly made the problem too complex for the model. Lastly, some user features were ineffective alone, but improved model performance when combined with other features. The results from this thesis combine to suggest a potential for incorporating user features to improve performance of hate speech detection. More research should be conducted to understand which features work well for different subtasks, continue to investigate other features and how these are most effectively utilized.

7.2 Future Work

Several studies have focused on improving automatic hate speech detection in the recent years, yet there is still need for more work in the area. This section presents suggestions for how the conducted research in this thesis can be further extended or improved upon, and work that the field of research could benefit from.

Feature Selection Approaches

In a study about the effect of feature selection, Zhang et al. (2018b) found that feature selection is able to select a very small set of the most predictive features and is therefore more powerful and achieves better results than models using carefully engineered features. By using a state-of-the-art feature selection process based on logistic regression with L1-regularization, it was possible to discard calculated "feature importance" scores below a threshold. A similar approach can be used to investigate the most important user features to enhance the performance of classification methods, instead of systematically incorporating the individual features and subsets as was done in the experiments in this thesis. In addition, it is possible to improve the feature selection through a more thorough investigation of feature correlation.

Exploring Other User Features

There is a great amount of information related to the users of Twitter that was not used in the experiments, that can be retrieved or derived from user behavior. This allows future researchers to continue the investigation of various user features in hate speech detection. Examples include considering the time of tweeting, investigations of relationships with other users, communication with other users, and what content users are exposed to. Related to the user profile, it can be also interesting to investigate user profile personalization to a larger degree. An example includes whether specific background colors are associated with specific

types of users. In addition, considering the amount of research conducted to find characteristic traits of hateful users, personal attackers and trollers, efforts should be made to utilize these features in automatic detection of hate speech.

Users of Various Online Platforms

Twitter was the platform of focus in the experiments in this thesis. However, Twitter is very different from other online communities, in terms of nature and functions. Therefore, the findings from the experiments cannot necessarily be generalized to represent users of other online communities. As mentioned, the gender of users is not directly retrievable through Twitter and may cause the inconsistent effects on performance. However, in other online communities, gender can be directly retrieved and may show more consistent effects. Future work should therefore consider to investigate the effects of user features in other internet communities and platforms as well. However, it should be mentioned that both the online communities and researchers developing methods for detection should consider the importance of perceiving freedom of speech.

User Features in Different Subtasks

Research in the field of hate speech detection often targets different subtasks, such as detecting bullying, racism or hate related to the refugee crisis. In the experiments in thesis, the subtasks were generalized to hate speech detection. However, as the incorporation of user features resulted in different effects based on the datasets tested on, this can imply that this generalization was not suitable. It is therefore important to find the most appropriate features for each subtask. Waseem et al. (2017) also implied that future research should aim to create a more robust understanding of when to use which features.

Bibliography

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760, Perth, Australia, 2017. International World Wide Web Conferences Steering Committee.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on neural networks*, 5(2), March 1994.

Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

Erin E. Buckels, Paul D. Trapnell, and Delroy L. Paulhus. Trolls just want to have fun. *Personality and individual Differences*, 67:97–102, 2014.

Peter Burnap and Matthew Leighton Williams. Hate speech, machine classification and statistical modelling of information flows on twitter: Interpretation and communication for policy decision making. *Policy and Internet*, 7(2):223–242, June 2015.

Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. Mean birds: Detecting aggression and bullying on twitter. In *Proceedings of the 2017 ACM on Web Science Conference*, pages 13–22. ACM, 2017.

Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. Detecting offensive language in social media to protect adolescent online safety. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom)*, pages 71–80, Amsterdam, Netherlands, 2012. Institute of Electrical and Electronics Engineers (IEEE).

Justin Cheng, Christian Danescu-Niculescu-Mizil, and Jure Leskovec. Antisocial behavior in online discussion communities. In *Proceedings of the Ninth International AAAI Conference on Web and Social Media*, pages 61–70, 2015.

Justin Cheng, Michael Bernstein, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. Anyone can become a troll: Causes of trolling behavior in online discussions. 2017.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. pages 512–516, 2017.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web*, pages 29–30, Florence, Italy, 2015. Association for Computing Machinery (ACM).

Paula Fortuna. Automatic detection of hate speech in text: an overview of the topic and dataset annotation with hierarchical classes. Msc, Faculdade De Engenharia Da Universidade Do Porto, Porto, Portugal, June 2017.

Lei Gao and Ruihong Huang. Detecting online hate speech using context aware models. *arXiv preprint arXiv:1710.07395*, October 2017.

Jennifer Golbeck, Zahra Ashktorab, Rashad O Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Paul Cheakalos, Alicia A Geller, Quint Gergory, Rajesh Kumar Gnanasekaran, et al. A large labeled corpus for online harassment research. In *Proceedings of the 2017 ACM on Web Science Confer-*

ence, pages 229–233, NY, USA, 2017. ACM.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

Claire Hardaker. Trolling in asynchronous computer-mediated communication: From user discussions to academic definitions, January 2010.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Irene Kwok and Yuzhou Wang. Locate the hate: Detecting tweets against blacks. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 1621–1622, Bellevue, Washington, 2013. AAAI Press.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196, Beijing, China, 2014. Journal of Machine Learning Research.

Veronica Lynn, Youngseo Son, Vivek Kulkarni, Niranjan Balasubramanian, and Andrew H. Schwartz. Human centered nlp with user-factor adaption. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1157–1166, 2017.

Yashar Mehdad and Joel R Tetreault. Do characters abuse more than words? In *Proceedings of the Special Interest Group on Discourse and Dialogue (SIGDIAL) 2016 Conference*, pages 299–303, Los Angeles, USA, 2016. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, pages 145–153, Montréal, Québec, Canada, 2016. International World Wide Web Conferences Steering Committee.

Briony J. Oates. *Researching information systems and computing.* Sage Publication, 2005.

Etienne Papegnies, Vincent Labatut, Richard Dufour, and Georges Linares. Impact of content features for automatic online abuse detection. *International Conference on Computational Linguistics and Intelligent Text Processing, Apr 2017, Budapest, Hungary. International Conference on Computational Linguistics and Intelligent Text Processing, 18, International Conference on Computational Linguistics and Intelligent Text Processing*, 2017.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135, Copenhagen, Denmark, 2017a. Association for Computational Linguistics (ACM).

John Pavlopoulos, Prodromos Malakasiotis, Juli Bakagianni, and Ion Androutsopoulos. Improved abusive comment moderation with user embeddings. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 51–55, Copenhagen, Denmark, 2017b. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empir-*

ical methods in natural language processing (EMNLP), pages 1532–1543, Doha, Qatar, 2014.

Adwait Ratnaparkhi. A simple introduction to maximum entropy models for natural language processing. *IRCS Technical Reports Series*, page 81, 1997.

Payam Refaeilzadeh, Lei Tang, and Huan Liu. On comparison of feature selection algorithms. In *Proceedings of AAAI workshop on evaluation methods for machine learning II*, volume 3, page 5, 2007.

David Robinson, Ziqi Zhang, and Jonathan Tepper. Hate speech detection on twitter: Feature engineering v.s. feature selection, April 2018.

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. In *Proceedings of NLP4CMC III: 3rd Workshop on Natural Language Processing for Computer-Mediated Communication*, volume 17 of *Bochumer Linguistische Arbeitsberichte*, pages 6–9, Bochum, sep 2016.

Anna Schmidt and Michael Wiegland. A survey on hate speech detection using natural language processing. *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, 59:1–10, April 2017.

Sara Sood, Judd Antin, and Elizabeth Churchill. Profanity use in online communities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1481–1490, Austin, Texas, USA, 2012a. Association for Computing Machinery (ACM).

Sara Owsley Sood, Judd Antin, and Elizabeth F Churchill. Using crowdsourcing to improve profanity detection. In *AAAI Spring Symposium: Wisdom of the Crowd*, volume 12, pages 69–74, Stanford, CA, United States, 2012b.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394, Uppsala, Sweden, 2010. Association for Computational Linguistics.

Filippos Karolos Ventirozos, Iraklis Varlamis, and George Tsatsaronis. Detecting aggressive behavior in discussion threads using text mining. 2017.

Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. Cursing in english on twitter. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 415–425. ACM, 2014.

Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL) Student Research Workshop*, pages 88–93, San Diego, California, June 2016. The Association for Compututational Linguistics.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84, Association for Computational Linguistics, 2017. Vancouver, Canada.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399. International World Wide Web Conferences Steering Committee, 2017.

Guang Xiang, Bin Fan, Ling Wang, Jason Hong, and Carolyn Rose. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1980–1984, Maui, HI, USA, 2012. Association for Computing Machinery (ACM).

Jun Yan. *Text Representation*, pages 3069–3072. Springer US, Boston, MA, 2009.

Ziqi Zhang, D Robinson, and J Tepper. Detecting hate speech on twitter using a convolution-gru based deep neural network, April 2018a.

Ziqi Zhang, David Robinson, and Jonathan Tepper. Hate speech detection on twitter: Feature engineering vs feature selection. April 2018b.

A. Stop Words

English Stopwords					
a	d	here	needn	should've	we
about	did	hers	needn't	so	were
above	didn	herself	no	some	weren
after	didn't	him	nor	such	weren't
again	do	himself	not	t	what
against	does	how	now	than	when
ain	doesn	i	o	that	where
all	doesn't	if	of	that'll	which
am	doing	in	off	the	while
an	don	into	on	their	who
and	don't	is	once	theirs	whom
any	down	isn	only	them	why
are	during	isn't	or	themselves	will
aren	each	it	other	then	with
aren't	few	its	our	there	wouldn
as	for	itself	ours	these	wouldn't
at	from	it's	ourselves	this	won
be	further	just	out	those	won't
because	had	ll	over	through	y
been	hadn	m	own	they	you
before	hadn't	ma	re	to	your
being	has	me	s	too	yours
below	hasn	mightn	same	under	yourself
between	hasn't	mightn't	shan	until	yourselves
both	have	more	shan't	up	you'd
but	haven	most	she	ve	you'll
by	haven't	mustn	she's	very	you're
can	having	mustn't	should	was	you've
couldn	he	my	shouldn	wasn	
couldn't	her	myself	shouldn't	wasn't	

Table 1: English stopwords from the NLTK library

German Stopwords					
aber	desselben	er	jedem	nur	vor
alle	demselben	ihn	jeden	ob	während
allem	dieselben	ihm	jeder	oder	war
allen	dasselbe	es	jedes	ohne	waren
aller	dazu	etwas	jene	sehr	warst
alles	dein	euer	jenem	sein	was
als	deine	eure	jener	seine	weg
also	deinem	eurem	jenes	seinem	weil
am	deinen	euren	jetzt	seinen	weiter
an	deiner	eurer	kann	seiner	welche
ander	deines	eures	kein	seines	welchem
andere	denn	für	keine	selbst	welchen
anderem	derer	gegen	keinem	sich	welcher
anderer	dessen	gewesen	keinen	sie	welches
anderes	dich	hab	keiner	ihnen	wenn
anderm	dir	habe	keines	sind	werde
andern	du	haben	können	so	werden
anderr	dies	hat	könnte	solche	wie
anders	diese	hatte	machen	solchem	wieder
auch	diesem	hatten	man	solchen	will
auf	diesen	hier	manche	solcher	wir
aus	dieser	hin	manchem	solches	wird
bei	dieses	hinter	manchen	soll	wirst
bin	dach	ich	mancher	sollte	wo
bis	dort	mich	manches	sondern	wollen
bist	durch	mir	mein	sonst	wollte
da	ein	ihr	meine	über	würde
damit	eine	ihre	meinem	um	würden
dann	einem	ihrem	meinen	und	zu
der	einen	ihren	meiner	uns	zum
den	einer	ihrer	meines	unsere	zur
des	eines	ihres	mit	unserem	zwar
dem	einig	euch	muss	unseren	zwischen
die	einige	im	musste	unser	
das	einigem	in	nach	unseres	
daß	einigen	indem	nicht	unter	
derselbe	einiger	ins	nichts	viel	
derselben	einiges	ist	noch	vom	
denselben	einmal	jede	nun	von	

Table 2: German stopwords from the NLTK library

Portuguese Stopwords					
a	essas	foram	isto	por	terá
à	esse	fôramos	já	qual	terão
ao	esses	forem	lhe	quando	terei
aos	esta	formos	lhes	que	teremos
aquela	está	fosse	mas	quem	teria
aquelas	estamos	fossem	mais	são	teriam
aquele	estão	fôssemos	me	se	teríamos
aqueles	estas	fui	mesmo	seja	teu
aquilo	estava	há	minha	sejam	teus
as	estavam	haja	minhas	sejamos	teve
às	estávamos	hajam	meu	sem	tinham
até	este	hajamos	meus	será	tinha
com	esteja	hão	muito	serão	tínhamos
como	estejam	havemos	na	serei	tive
da	estejamos	hei	não	seremos	tivemos
das	estes	houve	nas	seria	tiver
de	esteve	houvemos	nem	seriam	tivera
dela	estive	houver	no	seríamos	tiveram
delas	estivemos	houvera	nos	seu	tivéramos
dele	estiver	houverá	nós	seus	tiverem
deles	estivéramos	houveram	nossa	só	tivermos
depois	estivera	houvéramos	nossas	somos	tivesse
do	estiveram	houverão	nosso	sou	tivessem
dos	estiverem	houverei	nossos	sua	tivéssemos
ela	estivermos	houverem	num	suas	tu
ele	estivesse	houveremos	numa	também	tua
elas	estivessem	houveria	o	te	tuas
eles	estivéssemos	houveriam	os	tem	um
em	estou	houveríamos	ou	tém	uma
entre	eu	houvermos	para	temos	você
era	foi	houvesse	pela	tenha	vocês
eram	fomos	houvessem	pelas	tenham	vos
éramos	for	houvéssemos	pelo	tenhamos	
essa	fora	isso	pelos	tenho	

Table 3: Portuguese stopwords from the NLTK library

B. Experimental Results

	Precision	Recall	F1-score
n-grams	0.83	0.83	0.82
+ male	0.83	0.83	0.82
+ female	0.84	0.84	0.84
+ friends	0.83	0.83	0.82
+ followers	0.84	0.84	0.84
+ statuses	0.83	0.83	0.82
+ favourites	0.82	0.83	0.82
+ public lists	0.84	0.85	0.84
+ geo enabled	0.83	0.84	0.83
+ default profile	0.83	0.84	0.83
+ default image	0.83	0.83	0.82

Table 1: Average precision, recall and F1-score of the classifier with n-grams and individual user features on the dataset by Waseem and Hovy (2016)

	Precision	Recall	F1-score
n-grams	0.80	0.79	0.77
+ male	0.80	0.79	0.77
+ female	0.80	0.79	0.76
+ friends	0.80	0.79	0.77
+ followers	0.81	0.81	0.79
+ statuses	0.79	0.78	0.75
+ favourites	0.80	0.79	0.77
+ public lists	0.81	0.80	0.78
+ geo enabled	0.81	0.81	0.79
+ default profile	0.80	0.79	0.77
+ default image	0.80	0.79	0.77

Table 2: Average precision, recall and F1-score of the classifier with n-grams and individual user features on the dataset by Fortuna (2017)

	Precision	Recall	F1-score
n-grams	0.48	0.69	0.56
+ male	0.48	0.69	0.56
+ female	0.48	0.69	0.56
+ friends	0.80	0.72	0.64
+ followers	0.48	0.69	0.56
+ statuses	0.48	0.69	0.56
+ favourites	0.48	0.69	0.56
+ public lists	0.48	0.69	0.56
+ geo enabled	0.47	0.66	0.55
+ default profile	0.48	0.69	0.56
+ default image	0.48	0.69	0.56

Table 3: Average precision, recall and F1-score of the classifier with n-grams and individual user features on the dataset by Ross et al. (2016)