An evaluation of automated methods for hate detection

Oskar Lindholm
Abstract

An evaluation of automated methods for hate detection

Oskar Lindholm

Derogatory, foul, hateful and/or prejudiced comments or even threats directed at other individuals have become a common phenomenon in many digital environments. This is a problem that affects many levels of society, and being able to battle it is therefore of utmost importance. The large amount of data created every day creates a need for well working automatic methods for detecting this type of content. The subjective nature of hate, as well as the diversity of how it can be expressed, however, makes the creation of such methods somewhat difficult. In this thesis three different automated methods, developed by the Swedish defence research agency (FOI), for hate detection in texts have been evaluated. To aid in the evaluation of these methods and the disambiguation of hate as a concept, an attempt at defining hate based on psychology literature has also been made. The methods are tested using two different data sets: one handpicked set of comments aimed to test the variety in each methods hate detecting ability, as well as one in-the-wild-set aimed at testing the methods performances in a scenario of realistic application. The result shows a major difference of performance based on the set the methods are tested on. As well as the possible improvements that can be made to each method and the weaknesses of each approach, the result shows the difficulty of creating reliable methods for automated hate detection in general.
Sammanfattning

Acknowledgments

I would like to take the opportunity to thank everyone who has either been directly or indirectly involved in the fulfillment of this thesis.

First of all I would like to thank my supervisor Lisa Kaati (senior data scientist at FOI) not only for offering me the opportunity to take on this project, but also for her guidance, patience, and support which allowed me to enjoy the project and to work with high motivation. I would also like to thank Nazar Akrami (associate professor in psychology at Uppsala University) for his deep commitment with this thesis and especially for his guidance in the parts regarding the psychology of hate.

I would moreover like to thank the people of the data science group at FOI for helping me with everything from data collection and insight into the techniques this thesis touches on, to guidance in writing.

Last but not least I would like thank my reviewer Björn Victor for accepting the role and for his thorough evaluation and reviewing of my thesis.
5 Discussion

5.1 Ethical aspects

6 Conclusion

7 Future work
1 Introduction

The digital era has paved the way for a number of new ways to express ideas and opinions. Digital forums, comment sections, chat forums etc. have meant that it is now possible to express and discuss ideas and opinions whenever, with whomever and in whatever subject. That people can sit behind a screen and, often anonymously, discuss whatever subject they feel like has opened up a whole new dimension to the way we communicate [Kau12]. There is research showing that hidden behind a screen, people tend to loosen up and talk about things they might never have dared to otherwise, expressing themselves in ways they might never have done in a purely physical environment [Sul04].

The effects of this are both positive and negative. As well as people who loosen up and open up themselves have been shown to act more kindly and generously, it has also meant that some people, when online, express themselves in ways that would in most contexts never have been accepted [Sul04]. Derogatory, foul, hateful and/or prejudiced comments or even threats directed at other individuals have become a common phenomenon in many digital environments [AHS15]. Especially exposed to these kinds of comments are individuals with occupations that have a more public nature e.g. scientists, artists, journalists, and politicians etc [PH15]. In a study of online hate towards journalists made by Swedish defense research agency (FOI), about 32 percent of the journalists in the study were subjected to hate in at least one of more than 6,5 million comments analyzed [KO18]. Although an serious issue and even a threat against democracy when politicians and journalists quit because of their exposure to hate and threats [KO18, KAP18], very little research has been done regarding this type of hate in online environments. As a result, there are very little statistics showing how common it actually is [EKN18]. One issue is the lack of methods for analyzing the phenomenon in a larger context, working at a satisfactory level of performance. This can to some extent be derived from hate being a relatively subjective term with no single definition which therefore makes it tricky to analyze [DWMW17]. Several attempts to monitor hate speech in digital environments have been done using different automated approaches (e.g. [ISK18], [DWMW17], [NZHL15]) but the nature of hate complicates such approaches severely.

In this thesis, the advantages and shortcomings of automatic hate detection as a tool to better understand digital hate will be analyzed and evaluated. This will be done by analyzing and testing three methods for automatic hate detection using two different types of data sets and evaluating the result. The methods that will be tested have been developed by FOI, of which two base their assessments on dictionaries and one is based on machine learning. The analysis is focused, both on the problems that arise while trying to detect hate using automated methods, and what can be done to improve the
methods analyzed. The methods will be run and tested on data in Swedish. The reason for this is that the methods are adapted to work on text in Swedish.

The result from the evaluation showed a great difference in performance depending on what data the methods were applied on. It clearly shows the difference between testing the methods on data partitioned from the same set as the training data, and data that, although similar to the training data, is closer to what the methods would be applied on in a real-world application. For each method the performance was substantially worse when applied to data of the later type. Depending on the method that was used, however, varying types of hate and different nuances of hateful language were picked up. The result clearly shows the many difficulties in detecting hate using automatic methods. It also highlights the problems that comes with each method as well as the general problem of detecting hate that arises from the complexity of hate as a concept. Although the result mainly makes the problems of hate detection apparent, it also shows the strength of each method in distinguishing different aspects of a language and in detecting different kinds of hateful expressions. The result could potentially be used to improve each method or as a basis for developing new methods, trying to avoid the shortcomings with the methods shown in this thesis.

1.1 Background

FOI is one of the leading institutions in defense and security research in Europe and specializes in research in a wide variety of areas, ranging from the development of models, tools and methods for environmental adaptability of society, to IT security and human system interaction. In 2018, researchers at FOI conducted two studies regarding hate in online environments. One at the request of the Swedish Association of Local Authorities and Regions (SKL), investigating hate and threats directed towards local politicians in Sweden [KAP18]. The other one at the request by the Swedish Media Publisher’s Association (TU), investigating hate and threats directed at Swedish journalists [KO18]. For these studies two different methods of hate analysis were used. One dictionary-based algorithm to detect hateful or threatening texts from a number of comment sections and digital forums. For the other study, the comments were analyzed manually. Because of hate being an extremely complex term, something that can be expressed very differently (it is rarely as simple as an individual stating that he or she hates someone), using automatic methods has its drawbacks. Because of this, there was an interest at FOI for analyzing the shortcomings of the previously used methods closer and to investigate yet another alternative method for hate detection. It is this analysis that this thesis is focused on. The hope for such an analysis is to gain more knowledge about the short-
comings as well as the advantages of the different methods and to lay the groundwork for improvements.

The data used in the study will be taken from a Swedish context and the language used in all of the analyzed texts will be in Swedish. When such texts are used as examples in this report they will be freely translated and presented in English, in some cases along with the original text in Swedish.

1.1.1 Defining hate

In order to understand the complexity of hate detection using automatic methods, it is important to have a clear understanding of the problems of hate as a concept, the complexity of reaching a clear definition and the many ways in which it can be expressed.

In both psychology literature and earlier studies regarding automatic hate detection there have been several different definitions of hate and no clear consensus on what hate actually is [FHCJ18, Sel16]. The fact that that it generally is an issue to define hate means that defining what to look for when trying to detect hate, as well as comparing the result of earlier studies is somewhat complex. There is, however, within the frame of this study a need to make an interpretation based on the more general psychological perceptions of hate to define the focus of this study.

Generally in psychology literature hate is perceived as a strong, stable, negative emotional phenomenon and it is often described as being closely related to other negative emotions such as anger, disgust, contempt and the wish to seek revenge [FHCJ18]. It has, however, been argued whether hate actually is an emotion. Hate has, just to mention a few examples, been classified as a sentiment, an emotional attitude and a syndrome. One characteristic of hate that actually does separates it from many other feelings is that hate is usually perceived as being relatively stable over time [Ste03, FHCJ18]. To put a perspective on this, anger is usually a feeling directed at an individual based on an action, by the target, that the angry person has perceived as unjust or unfair [FR07]. Research on anger has also shown that anger, to some extent, can be seen as a means of trying to force the target to change the behavior that caused the anger. What this means is that, while anger often fades away or disappears by time, sometimes because the individual that the anger is aimed at changes his or her behavior e.g. by apologizing, hate is steadier. Hate is usually based on a stable perception of a person or group, their malevolent and immoral nature, their malicious intent and their inability to change [RMR05, FHCJ18]. This often results in a will to avoid, punish, hurt or even eliminate or destroy the hate target. This is, furthermore, often based on the target being perceived as a threat or as of having treated the hater unfairly.
Although we sometimes hate persons that we know, it can be argued that the most distinguishing characteristic of hate is that to hate someone, no relationship needs to exist between the hater and the target of hate [FHCJ18]. Hate can be, and is often, directed at a person or a group, based on a perceived attribute of this group or individual or on something they represent e.g. power, a certain identity or behavior. The hater usually feels or has felt threatened, powerless, humiliated etc. by the group or individual that the hate is directed at. However, it might not have been because of something that the hater has been exposed to directly. It might not even be because of something that the target has actually done. What the target represent is enough [FHCJ18]. An example of this is hate against homosexuals. People can hate homosexuals because they perceive homosexuality to defy the laws of nature or because they perceive it to be sinful. For this to be, it often doesn’t matter whether a person hating homosexuals knows or has ever really met someone that is known to them to be homosexual. It could also be so that someone hates Muslims because they perceive them as a threat, thinking that Muslims are representatives of Islamic terrorists, that all Muslims must be terrorists, despite the fact that he or she has never talked to a single person who is a Muslim.

That hate often is based on certain attributes that the hater perceives an individual or a group to have or represent, also means that hate is often seen as being closely related to prejudice [DGP05]. Hate based on these kinds of attributes and prejudices expressed in a public context can be and is often classified as hate speech [Sel16]. In turn, hate speech can, depending on the laws of the country from which a definition is drawn, be classified as a hate crime. The definition of both hate speech and hate crimes can sometimes vary depending on several factors (e.g., as mentioned, laws of different countries). However hate crimes can in a general sense be defined as a crime directed at people based on them being or being perceived as affiliated to a certain group of people e.g. based on ethnicity, sex, sexual orientation or religion [Pet03, Ger03]. In Sweden, such crimes include attacking the identity and dignity of an individual (e.g. sexual orientation or religious beliefs), or a group (e.g. immigrants, Muslims, transsexuals). Included in hate crimes is also actions or baiting to actions intended to harm individuals because of their belonging, or perceived belonging, to such groups. Hate speech can, in its turn, from this definition be defined as a verbal attack on an individual or group based on such attributes or characteristics (perceived or real).

Most earlier studies regarding automatic detection of hate have been focused on hate speech [DWMW17, DZM+15, NZHL15]. The main focus of the methods presented in this study is not to detect hate speech exclusively. The methods can very well be used to detect milder forms of offensive language as well as hate that is not based on the attributes mentioned in the previous section.

The definition used in this thesis is therefore somewhat broader than what has been used in many other studies. The aim is to also include a more thorough definition of
how hate is being expressed, not just by defining the possible basis for expressing it (i.e. the groups or attributes towards which hate can potentially be aimed). Hate has in this thesis been defined as:

*To, based on personal attributes or characteristics such as race, sex, religion, occupation or political views, directly insult or show aggression, disgust or malice against a person.*

This definition is partly based the definition of hate and hate speech given earlier, but also on a definition given in the annotation tool used to collect data for the study.

It should furthermore be noted that the kind of target at which hate is aimed can differ. Hate can be directed more generally at a certain group, e.g. immigrants, homosexuals, Muslims, politicians etc. by just referring to the group itself e.g. "Fucking immigrants, they are destroying our country". It can also be so that hate is being directed at a certain individual, which in turn can be based both on that individual belonging, or being perceived to belong to a certain group e.g. "@person is a fucking nigger" or just that the hater hates that individual in particular e.g. "@person is so fucking ugly, I hope he dies". The main focus when it comes to catching hateful content will in this thesis be the latter category i.e. hate directed at certain individuals, based on the definition given for this thesis.

### 1.1.2 The complexity of detecting hate

When it comes to actually detecting hate using automated methods the complexity is twofold. It lies both within the subjective nature of hate (what one person considers hateful might not be considered hateful by someone else) but also in the subtle way that it is sometimes being expressed. As earlier discussed, hate is a difficult term to define in a straightforward way, and contained within hate is a range of different emotions. Even with the definition given earlier, defining how hate is being expressed explicitly is an extremely hard task. Hate can, as well as being based on a number of attributes (e.g. race and gender), be expressed in a number of ways ranging from foul nicknames to threats of life or assault. An individual expressing hate can sometimes perceives the target as a threat and therefore express hate against that target in a threatening way. Among the most difficult texts to catch are the ones that could be said to contain threats. A threat for example, could very well be expressed using words that are not threatening nor offensive on their own. For example,

"The next time I see you and have my knife at hand who knows what will happen".
Such a sentence is for a human very easy to understand as hateful or as a threat. For an algorithm, however, there is next to nothing that distinguishes this comment as hateful or as a threat. "Knife" on its own is not nearly enough to point to as a sign of hate, it could very well be that a person using this word is talking about cooking. Nor does it say in the sentence what the knife will be used for or how. It is this subtlety that can be extremely hard to catch.

When searching through a text for hate directed towards a certain person or when trying to detect towards whom hate in a comment is being directed, there are even several further issues arising. One example is when two or more persons are mentioned but hate is only directed towards one of these subjects:

"How the fuck did @person1 get an erection when @person2 was present? She looks disgusting, cadaverous and just disgusting. Did @person1 take a Viagra and did he have a paper bag with him?".

The hateful content in such a sentence is only directed at @person2 not towards @person1. Furthermore, @person2 is not directly the subject of the hateful words used but instead referenced to using a pronoun (she).

Furthermore as hate is closely related to anger, a person that feels anger can sometimes express him or herself in a way that sounds hateful but that might not be so. It can, for example, be so that an individual writes a comment stating to hate a person while the actual feeling might only be anger:

"I hate @person for what he did to me"

It is however impossible to know the exact feelings of a person writing such a comment. As discussed earlier, hate is related or even includes a relatively wide range of emotions. This also means that hate is often being expressed as, or by using these emotions to describe hate towards an individual. Such an example is given in the sentence below where disgust is used:

"Due to the potentially sensitive nature of some of the terms used in the sentence below they have been replaced with @property1 and @property2"
@person is so fucking disgusting, not only is he @property1 but a @property2 as well.

The complexity that this brings means that very little can be done in order to take the potential feelings behind an expression into account. The basis for classification of hate in text in this thesis is therefore based on how individuals express themselves in writing alone.

1.1.3 Hate from a legal and democratic perspective

As mentioned earlier, hate can when expressed in certain ways be classified as a crime. This was also explained to depend on the laws from which a definition is drawn. A country that for example is relatively liberal when it comes to what can be expressed is the United States [Sel16]. Why this differs usually depends on how liberal the laws are regarding freedom of speech in a country. Even though it has been argued that to restrict hate speech is to violate the foundations of democracy, that it is a restriction of the freedom of speech [Hei16], hate speech (and hate in general) has been, and is, a problem that in itself can be a threat against democracy. A report on threats and violence against occupational groups that are important in a democracy (e.g. journalists and politicians) by the Swedish National Council for Crime Prevention (BRÅ) [PH15], shows that many people in these occupational groups are often being exposed to both threats and other types of harassment. A survey regarding the safety of politicians also made by BRÅ [Fre17] shows that it is not unusual for politicians to either think about resigning or to actually resign because of hate and threats that are being directed at them.

1.2 Aim

In order to shed light on some of the problems and advantages of using automatic methods to detect hate, this thesis aims to analyze and evaluate the advantages and shortcomings of three technical solutions for detection of hate expressed in digital environments. Two dictionary-based algorithms, and one based on machine learning trough logistic regression. The methods will be tested using real data from Swedish online discussion forums and comment sections. The methods will first be tested on a smaller set of annotated data, i.e. texts marked as hateful or non hateful, and then used to analyze a larger data set that has not been annotated. The aim of this second stage is mainly to assess how well the chosen method performs in the wild and if the assumptions made in the first stage are correct. This will be done by annotating a small sample from this larger data set manually and compare it to the classifications made by the different methods.
The methods will be described and analyzed from a technical perspective i.e. why a certain method works better or worse and why some methods tend to work better on some texts than on others. The aim is furthermore to, as well as pointing out the weaknesses each method carries, to find possible improvements.

1.3 Delimitations

Although the aim of analyzing and evaluating the hate detection methods in this study, is partly to provide suggestions of improvements to each method, the improvements will not be tested nor evaluated. Furthermore, the direct implementation of the methods that will be tested will not be shown or discussed in deeper technical details e.g. language and modules used in the actual implementation\textsuperscript{2}. This also means that possible improvements suggested after the analysis is performed will not be implemented either. The possible improvements will simply be based on the analysis made in the study and on how related methods work. Furthermore, this thesis will only concern hate detection in text i.e. comments made in, for example, online forums and comments sections of digital newspapers. Detection of symbols, memes or symbolic expressions of hate will not be discussed in this paper.\textsuperscript{3}

As mentioned earlier the methods that will be analyzed in this thesis will only be used on data written in Swedish. The reason for this being that the methods are adapted to work on text in Swedish. It is, however, possible to adapt the way these methods work to other languages but the actual implementation of these methods are highly dependent on what language it is designed to work on. It is not the aim of this thesis to describe the adaption of the methods to work on other languages and noting related to such adaptions will be discussed nor described in this thesis. All analysis will therefore will take its basis in how well the methods work when analyzing text in Swedish.

\textsuperscript{2}For a more technical description of the implementation of the lexical based methods than the one given in this thesis see \cite{KAP18} and \cite{PKA18}

\textsuperscript{3}For more information regarding hate symbols see: \textit{Det vita hatet: radikal nationalism i digitala miljöer} 2017 By Lisa Kaati et al Available at: https://www.foi.se/rapportsammanfattning?reportNo=FOI-R--4463--SE

8
2 Method

The analysis in this thesis will be made on three different methods: two lexical based approaches and one machine learning based approach.

The method used to test and evaluate the hate detection methods in this study can be split into two stages as seen in Figure 1 and 2: testing and in the wild. In the testing stage, the methods will be run on a sample of annotated data to determine how well they can actually detect hate. This data set will consist of 50/50 hate and non hate texts. After this a much larger, unannotated data set will be used to evaluate the performance in the wild. A smaller sample of around 600 comments will then be taken from the classifications made in this stage. The idea behind this later stage is to evaluate performance on texts that have not been specifically chosen to use as test data.

Figure 1 A visualization of the work flow used in the test part of the study
2 Method

2.1 Measurements

In order to assess how well the different methods work, as well as to create an easy way of comparison between them (both in the testing stage and in the wild), four measures, as well as a confusion matrix will be used as a basis from which the performance of each method can be calculated and compared.

A confusion matrix divides the classifications made by a method based on the correctness of each classification. From this, each of the four measurements can be calculated based on the ratio between different types of classifications. A confusion matrix consists of a two-dimensional matrix that in one dimension is indexed by the true class of an object, and in the other by the class assigned by the classifier. It can be used when there are multiple classes but will in this study be used with two classes i.e. hate and not hate. This can be said to be a special case of the confusion matrix and the dimensions in such a matrix are indexed by positive (hate) and negative (not hate). A standardized two dimensional confusion matrix is shown in Figure 3. In Figure 4 a confusion matrix labeled in accordance with this study i.e. the output/predictions made by the methods that will be analyzed is shown. In the later figure, yes means that a text has been classified as containing hate and no means that a text has been classified as non-hatefull. The confusion matrix allows the evaluation to be made based on four types of predictions, True Positive, true negative, false positive and false negative. The first two types signifies correct classifications and the two later incorrect classifications. True Positive
(TP) means the method has predicted an observation as positive (i.e. in this case as hate) that truly is positive. True negative (TN) means the method has predicted an observation as negative (i.e. in this case as not hate) that truly is negative. False positive (FP) means the method has predicted an observation as positive that actually is negative and False negative (FN) means the method has predicted an observation as negative that is actually positive. When defining formulas based on these types of classifications the abbreviations presented here will be used.

Figure 3  A standardize confusion matrix

<table>
<thead>
<tr>
<th>Actual class</th>
<th>Assigned class</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Positive</td>
<td>Negative</td>
</tr>
<tr>
<td>Positive</td>
<td>TP</td>
<td>FN</td>
</tr>
<tr>
<td>Negative</td>
<td>FP</td>
<td>TN</td>
</tr>
</tbody>
</table>

Figure 4  The confusion matrix used in this study

<table>
<thead>
<tr>
<th>Actually</th>
<th>Classified Yes</th>
<th>Classified No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actually Yes</td>
<td>TP</td>
<td>FN</td>
</tr>
<tr>
<td>Actually No</td>
<td>FP</td>
<td>TN</td>
</tr>
</tbody>
</table>
Based on these four classification types, different measures of classification performance can be derived. In this study four such measures will be used: Accuracy, Precision, Recall, and Specificity. Accuracy measures how well a method actually can predict the true class of all observations [SW11] i.e. the ratio of the sum of True Positives and true negatives, and the total number of observations. Precision is used to measure how well a method can predict True Positives and is defined as the ratio of True Positives and the total number of True Positives. Recall measures how well a method works in finding positives correctly i.e. the ratio of predicted positives and the total number of positives in the set. Specificity measures how well a method predicts negatives correctly i.e. the ratio of predicted negatives and the total number of negatives in the test set. The formulas for each measure can be seen in Figure 5.

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1) \\
\text{Precision} = \frac{TP}{TP + FP} \quad (2) \\
\text{Recall} = \frac{TP}{TP + FN} \quad (3) \\
\text{Specificity} = \frac{TN}{TN + FP} \quad (4)
\]

Figure 5 The formulas used to calculate the prediction performance of each method.

In order to use these measurements as a basis for evaluation and to facilitate a comparison between the methods presented in this thesis and similar methods, there is a need for a clear goal in relation to how well the methods ideally should perform. When evaluating machine learning classifiers a common practice is to use a baseline classifier (as done by Wester et al. [WØVH16]). A baseline classifier is a naive classifier that is expected to perform worse than the classifier that is being developed. The baseline classifier provides a baseline from which a minimum of how well the other classifiers can be expected to perform can be derived. As only one machine learning method is used in this case it could potentially be misleading to use a machine learning classifier as a base for judging the performance of the non machine-learning-based classifiers. Because of this, no baseline classifier will be used. The basis for how well the methods are required to perform will instead be how well the methods perform in relation to each other and on how well similar methods have worked.

Furthermore, it is important not only to know how well a method should be expected to perform but how it is expected to perform. What this means is that, depending on the context in which a method is used and what it is used for, different measures are more important than others. Recall for example plays a major when using classifiers to predict if a patient has a disease or not, where the main goal is to identify all real positive
cases [Pow08]. It is also important to consider what is being measured, as using the wrong measure in certain situations can be extremely misleading. An example is when it is crucial to be able to identify a class that only makes up a small percentage of the entire population. It could, for example, be so that one wishes to separate terrorists boarding a plane from regular passengers. Let’s for the sake of the example say that out of the entire population (all passengers boarding a plane) 0.01% are terrorists. Even a method that classifies every passenger as a non-terrorist (negative) would, when looking at its accuracy, recall and precision show a really well working approach as a large majority (99.99) percent of the population would be correctly identified. Looking at the specificity in such a case tells a different story, showing a rate of roughly 0%. It is in such a case easy to see how misleading these measures can be without considering what it actually is that one wishes to achieve in identifying the different classes. It is because of this, as well as a wish to compare and evaluate the methods based on a broad basis, that the four measures earlier presented are used in this study.

As automated hate detection can potentially be used to assess the threat against a certain individual it can be problematic, both to over and to underestimate hate and threats directed at such an individual. It is therefore important to consider how well a classifier should work when identifying both True Positives (hateful texts) and true negatives (non-hateful texts). In this thesis the foremost aim is, however, not to overestimate hate in a data set. It is therefore important to consider how well a method is able to recognize non-hateful texts, not creating an overestimation of comments predicted to contain hate that does not (i.e. false positives). The most important measures can therefore in relation to this be said to be precision and specificity. All measures presented as well as the numbers shown in the confusion matrix will, however, be taken into account when evaluating each method.

### 2.2 Analyzed methods

The methods chosen to be analyzed have, as mentioned earlier, been developed by researchers at FOI and some have to a certain extent been used in earlier studies. The methods will here be presented, how they work and operate, but not exactly how they are implemented.

#### 2.2.1 Lexical sliding window

The lexical sliding window method is arguably the simplest method used in this study. In order to detect hate it uses a list of words and a list of names to search for in a text. The list of words needs to be a list of words classified as hateful. The list of names
should contain names of individuals that the hate detection is aimed at. The method works by searching texts for names from the list where, in the vicinity of that name, one or more words from the word lists are used. To control the precision of this method the context within which the mentioning of names and words are allowed to occur in order to be classified as hate is limited within an n-gram sized window (i.e. a window the size of a set number of characters/symbols or words). Using a sliding window to narrow the context, making it span around two or three words around a target’s name means that the probability of that term referring to something or someone else than that target is lowered. Below is a visualization of how this works with a sliding window of size 3. The name of the target and the hate word is shown in bold.

I hope [person gets murdered] and he’s wife gets raped.

In such a comment murdered is directed at @person while raped is directed at his wife. Using the sliding window in combination with this simple lexical method means that text only containing hateful words that are directed at a person whose name is not on the list used, are not classified as hateful. In this study, the size used for the sliding window is determined by the size of the currently analyzed sentence. This size has been chosen on the basis that using a smaller window has earlier shown poor results, meaning that the algorithm discovered almost no hateful texts at all.

2.2.2 Dependency reasoning

The dependency reasoning method can be said to be a further development of the lexical sliding window. The input data needed for it to operate is, just as for the sliding window method, a list of names, a hate dictionary (a list of hate phrases and words) and the data to analyze. It operates, however, in a significantly more elaborate way, utilizing a combination of methods from natural language processing (NLP) and automated reasoning (AR). Additionally it is for this method possible to specify the gender corresponding to a name of a target person in the list of names, as well as specifying whether a phrase in the hate dictionary is directed only at the agent or patient of a sentence (for an explanation of agent and patient see the box below).
2 Method

Agent/Patient

Agent
An agent is the initiator (in this case the individual) of an action/the verb in a sentence i.e. "Billy is painting his house" where Billy is the agent of painting. In a more hateful sentence this could, for example, look like "Billy disgusts me." Where Billy is the agent of the verb disgusts.

Patient
The Patient is the one (in this case the individual) in a sentence that undergoes an action or change i.e. I am going to kill Billy, where Billy is the patient of the verb kill.

The process that this method uses to detect hate in text can be separated into five steps:

Step 1: Segmentation and tagging of words and punctuation marks.
Step 2: Dependency parsing (creating a parse tree from tagged sentences).
Step 3: Processing of target and non target nodes.
Step 4: Processing of sentiment nodes.
Step 5: Reasoning.
Step 1. Segmentation and tagging

The first operations performed on a given text is to segment it into words and punctuation symbols (i.e. question marks, commas, and interpunct etc.) Each word and symbol is then tagged with its most likely canonical form and PoS (Part of Speech) category e.g. see Figure 6

<table>
<thead>
<tr>
<th>Leon May</th>
<th>är</th>
<th>helt</th>
<th>galen</th>
<th>,</th>
<th>jag</th>
<th>hatar</th>
<th>honom</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROPN</td>
<td>PROPN</td>
<td>VERB</td>
<td>ADV</td>
<td>ADJ</td>
<td>PUNCT</td>
<td>PRON</td>
<td>VERB</td>
</tr>
</tbody>
</table>

An example of how this could work in English is (note that these sentences are not translations of each other)

<table>
<thead>
<tr>
<th>Leon May</th>
<th>is</th>
<th>mad</th>
<th>.</th>
<th>He</th>
<th>probably</th>
<th>sniffs</th>
<th>glue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leon May</td>
<td>be</td>
<td>mad</td>
<td>.</td>
<td>He</td>
<td>probably</td>
<td>sniff</td>
<td>glue</td>
</tr>
<tr>
<td>PROPN</td>
<td>PROPN</td>
<td>AUX</td>
<td>NOUN</td>
<td>PUNCT</td>
<td>PRON</td>
<td>ADV</td>
<td>VERB</td>
</tr>
</tbody>
</table>

Where:

- ADJ = Adjective
- AUX = Auxiliary verb
- ADV = Adverb
- NOUN = Noun
- PRON = Pronoun
- PROPN = Proper noun
- PUNCT = Punctuation mark
- VERB = Verb

Figure 6 An example of how the segmentation and tagging of words in a sentence is performed in the first step.
Step 2. Dependency parsing

The second step is used to turn the tagged sentences into dependency-based parse trees. A parse tree is a rooted tree that, based on some context-free grammar represents a string's syntactic structure [SW16]. A dependency-based parse tree is, in turn, a parse tree that based on dependency grammar, structures a sentence solely based on the words in a sentence and their dependencies [JM18]. A dependency can be said to be a connection of linguistic units, e.g. words, by directed links. At the structural center of a sentence, i.e. the root of the tree, is the finite verb. In a sentence all syntactic units are connected to this verb either directly or indirectly by links i.e. the dependencies.

![Figure 7](image_url) The dependency tree built after the second step

In Figure 7 is the sentence given in the example from step one, shown as is would look in this step. The text has been split into two dependency-based parse trees, one for each sentence.

Step 3. Processing of target and non target nodes

In the parse tree created in the previous step, personal nodes i.e. the ones marked as representing a last and first name are, if possible, merged into one node. These nodes are then (if the name matches a target) tagged with the full name and the gender of that target. This node is then marked as a target node which can be seen in Figure 8.

The gender of a target is later used to identify whether personal pronouns (i.e. he, him, she, her) are used in other parts of the text to refer to a person earlier mentioned by name (as seen in the example where "he" is used in the second sentence to refer to Leon May). This gender identification is done based on the list of targets (given that the gender is defined in relation to the name). If the gender hasn’t been defined in the list of targets the gender classification is done based on lookup in a list of 350,000 first names. For the process of tagging targets to work, it is not required for the actual name of a target to be specified in the target list, a mere nickname works. If a person name node does
2 Method

Leon May

Figure 8 The dependency tree after the third step with the tagged target node

not match a name or nickname of a target person it is marked as a non-target person node. This is in a later stage used to aid in the disambiguation of towards whom hate in a sentence is directed.

Step 4. Processing of sentiment nodes

Based on the hate dictionary the nodes containing words expressing hate and threats are found and marked (see Figure 9). This method also allows identification of different kinds of hate based on the words used in a text e.g. threats (e.g. kill, rape, attack, shoot), hate directed at minorities (e.g. nigger, homo, Jewish fucker), sexist hate (e.g. bitch, whore, bimbo). As the aim of this thesis is to analyze the detection of hate directed at an individual in general, this feature will not be part of the analysis. In the figure shown below, hate words have therefore not been marked as belonging to a category other than hate as a whole.

Figure 9 The dependency tree after the fourth step with the identified hate nodes tagged

As seen in Figure 9 this method allows for the detection of multi-word phrases even when the words making up such phrases are separated in the text. Nodes that correspond to words that are part of a multi-word phrase (e.g. "sniff glue") in the hate dictionary are marked as corresponding to the full phrase even if the words making up such hate phrases are not used next to each other in the text. This is, however, only done if the remaining part of a phrase can be found in the same subtree as the first part. This means that all nodes in the full phrase need to be dependent on each other, indicating that the
sentence actually contains the phrase used as a whole, not just the individual words used separately. It could, for example, be so that the phrase "deserve to die" is included in the dictionary and there is a sentence:

"@person1 deserves a warning but I hope @person2 dies".

This sentence would without checking that "deserve" and "die" are directly connected to the same parent node be interpreted as if they belong to the same phrase and be marked as hate directed at @person1 using the phrase deserves to die.

**Step 5. Reasoning**

The last step performed is designed to determine towards whom a hate phrase is directed. This is done in two phases by using the parse tree and the sequential order of words in each sentence, the steps taken are.

1.) The first phase utilizes reasoning by forward chaining. Forward chaining takes its basis in available data, and by using inference rules extracts more data with the aim of reaching a set of goals. A set of rules are set up that, if a case where an IF-clause from the rule sets is found to be true, the statement given as the THEN-clause can be inferred to be true and additional data can be added to the data from which a conclusion can be drawn. Here the focus lies on, by coreference resolution, attempting to identify to which name a personal pronoun is referring (both target and non-target). This is done by associating the personal pronouns with a person node of matching gender (see Figure 10).

![Figure 10](image-url)

*Figure 10* The dependency tree after the first part of the fifth and final step. Here the pronoun "he" has by reasoning trough forward chaining been identified as to referring to Leon May.
In the second phase backward chaining is used. Backward chaining can be said to work the other way around, compared to forward chaining. Backward chaining starts with set up goals and the data that the forward chaining draws its conclusions. The procedure is then to try and prove the rules needed in order to infer the truth of a goal. This is done starting at the THEN-clause of the rules and by checking if the IF-clause preceding it is know to be true the goal to be reached can be inferred. Additional data can hence be added to the data from which a conclusion is drawn. Reasoning by backward chaining is in this step used to find hate expression nodes directed against a target person node. The basis for this (if defined) is the agent/patient restrictions of hate verbs in the dictionary and dependencies such as a hate adjective node depending on a person node. This final step in addition to the data conclusion drawn in the previous step can be seen in Figure 11.

![Figure 11](image-url)

Figure 11 The dependency tree after the second step of the fifth and final step. Here the target of the hateful content has been identified through reasoning by forward chaining.

In the case of the example, this method would finally result in registration of two occurrences of hate towards Leon May such as:

<table>
<thead>
<tr>
<th>Hate</th>
<th>Target</th>
<th>Hate phrase</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>&quot;Leon May&quot;</td>
<td>&quot;Mad&quot;</td>
<td>Leon May is mad. He probably sniffs glue</td>
</tr>
<tr>
<td>Yes</td>
<td>&quot;Leon May&quot;</td>
<td>&quot;Sniffs glue&quot;</td>
<td>Leon May is mad. He probably sniffs glue</td>
</tr>
</tbody>
</table>

2.2.3 Bag of n-grams classifier

The bag of n-grams method uses several steps to obtain information contained in a text corpus in order to extract data to train a machine learning model. The bag of n-grams is the first step in this process.

A bag of n-grams model is a simplified representation of a text where the text is broken down into tokens i.e. elements or parts, of which a text is built. Each token consists of
an n-gram i.e a contiguous multiset of (depending on the application) letters, word pairs or phrases etc. Each text in this model is represented as an unordered multiset of tokens. The only thing kept after breaking down a text into a bag of n-grams representation, is the frequency in which each token occurs [JM18]. The size of a token in the method used here is at maximum a bigram i.e. an n-gram of size two, in this case words, and the at minimum a unigram i.e. an n-gram of size one.

As a basis for explaining how this method works, a bag of n-grams model where the size of the tokens are always the size of a word (also called a bag of words) will be explained along with the preceding steps used in this method. Below is an example of how the bag of n-grams (as mentioned, in the case of the example a bag of words) is created.

*Strike as I struck the foe, Strike as I would Have struck those tyrants, Strike deep as my curse. Strike, and but once*

Below the frequency with which each word occurs within the text given above is listed.

\[
\begin{align*}
\text{strike} & = 4 & \text{those} & = 1 \\
\text{as} & = 3 & \text{tyrants} & = 1 \\
\text{I} & = 2 & \text{deep} & = 1 \\
\text{struck} & = 1 & \text{my} & = 1 \\
\text{the} & = 1 & \text{curse} & = 1 \\
\text{foe} & = 1 & \text{and} & = 1 \\
\text{would} & = 1 & \text{but} & = 1 \\
\text{have} & = 1 & \text{once} & = 1
\end{align*}
\]

For a given corpus, a vocabulary containing all tokens is created in the form of a one-dimensional matrix, a vector, where each token is represented as an element in the vector. Each text contained within the corpus can then be represented by such a vector where the frequency of each token is given. For a given text this means that there are usually a number of elements in its vector representation that are 0, due to a token not being present in the text. An example of how a text is broken down, with a vocabulary based on the first text, is given below:

*I struck my foe as I once would Have struck those tyrants*

would be represented as the vector:

\[
[0 \ 1 \ 2 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1]
\]

---

4Taken from "Marino Faliero, Doge of Venice" by Lord Byron
In order to create a representation that is more useful for training the machine learning model, a Tf-idf is in this method used to create a weighted matrix. Tf-idf is an abbreviation where $Tf$ stands for term frequency and $idf$ for inverse document frequency. Tf-idf is a way to reflect how important a word is in a corpus or collection of documents, not only depending on the frequency with which a certain word occurs. What this means is that very common terms that are not unique to a certain type of texts such as: is, and, but, I etc. in a Tf-idf representation reflects the low importance of such terms in relation to distinguishing a certain type of text (e.g. a hateful one) from another (e.g. a non-hateful one). How the Tf-idf is calculated can be seen in figure 12.

\[
Tf = \frac{\text{Number of times a term appears in a text}}{\text{Total number of terms in a text}}
\]  

(5)

\[
idf = \log\left(\frac{\text{Number of texts}}{\text{Number of texts a certain term appears in}}\right)
\]  

(6)

\[
Tf \cdot idf = \frac{\text{Number of times a term appears in a text}}{\text{Total number of terms in a text}} \times \log\left(\frac{\text{Number of texts}}{\text{Number of texts a certain term appears in}}\right)
\]  

(7)

**Figure 12** Equations used to calculate the term frequency–inverse document frequency of a given corpus

For example, imagine a corpus of 100 000 texts that the texts given above are a part of, and the word "strike" appears 100 times, the tf-idf for "strike" within the first text, within that corpus would then be.

\[
tf idf = \frac{4}{16} \cdot \log\left(\frac{100000}{100}\right) = 0,75
\]  

(8)

The value 0,75 shows that the word "strike" is a relatively significant one for the kind of texts in the example as it, in relation to the entire corpus is relatively uncommon, but not in this particular text.
As training data for the machine learning model, the bag of n-gram representation of each text is given as a tf-idf representation in the form of a vector. This could for the text given in the second example above, based on the tf-idf calculation example given above (see Equation 8), look like.

\[
\begin{bmatrix}
0 & 0.08 & 0.3 & 0.03 & 0 & 0.2 & 0.03 & 0.2 & 0.1 & 0.08 & 0 & 0.1 & 0 & 0 & 0 & 0.01
\end{bmatrix}
\]

For the final stage of this method, the data produced by the procedure explained above is used to train a machine learning model. The model chosen to be used is logistic regression. This choice was based upon minor testing of how some commonly used machine learning classifiers (i.e. random forest, gradient boosting trees, Naive Bayes and logistic regression) performed with the training data. Among these the logistic regression model performed the best. That logistic regression turned out to be the best classifier in this context can most likely be derived from its ability to handle sparse vectors i.e. vectors containing a large number of zeros. Due to a large vocabulary produced for the text corpus used as training data, and texts within this corpus tending to contain a small percentage of the words in this vocabulary, the training data contains a large number of sparse vectors. As the purpose of this thesis is to evaluate the method explained in this section (along with two others), the preceding work of testing and evaluating the underlying classifier in relation to other machine learning methods will not be described.
2.3 Data

The data set used for this study can be divided into three areas of usage, input data, data for first stage testing and data for testing the methods in the wild. Included in the input data is training data for the machine learning model and lists of names and words used by the dictionary based algorithms. The different data sets and where they are used can be seen in figure 13.

Figure 13 The work flow of the test part of the study showing where the different types of data is used

2.3.1 Sources and Collection

In the first stage of testing, the three methods will be used upon a smaller annotated set of texts. The data used at this stage will, in order to create a realistic test situation, consist of user-generated comments. What this means is that the data used for testing at this stage will be taken from similar sources as the one from which data will be collected.
Method

to test the methods in the wild. This is also the case when it comes to the data used to
train the machine learning model. The test and training data will be collected from a
number of Swedish online forums and comment sections e.g. Samhällsnytt (a Swedish
alternative media website for spreading and discussion of news), Familjeliv (a Swedish
web forum focused on family life) and the comment sections of the Nordic Resistance
movements (NMR) website, Nordfront.se. When this study is being performed this data
has already been collected and used in earlier studies.

The chosen source of data for testing the methods in the wild is Flashback.se, a Swedish
web-forum focused on, in the words of the site itself: "Freedom of speech, opinion, and
independent thoughts". It is one of the largest web forums in Sweden and has around
1.1 million registered users [KO18]. Flashback is divided upon close to 300 subforums
and with the motto ‘Freedom of speech for real’, Flashback allows their users to discuss
subjects ranging from travels and sports to drugs and reviewing of prostitutes. The
choice of Flashback is based upon its diversity of subjects and type of users as opposed
to, for example, an alt right forum that mostly attracts user with an alt right political
opinion. Flashback also has the advantage of being non context based i.e. compared to
comment sections of news sites, it is not bound to any particular content on the site (e.g.
an article). In order to create a more diverse data set the data collected from Flashback
will not be from a particular forum but from the entire website. It will furthermore be
collected with the criteria that the texts to be included in the in-the-wild-data set must
contain at least one of the names from the name lists produced for the lexically based
methods, and that they need to be written in 2018.

The list of names as well as the list of words used by the lexically based methods will be
collected using a tool that utilizes a word2vec model. A word2vec model uses a neural
network to reconstruct a linguistic context of words from a given corpus. To the neural
network, a corpus is given, broken down into vectors based on a vocabulary (much like
the ones produced in the first stage of the bag of n-grams model). The neural network
is then trained to reconstruct the linguistic contexts of the words in this corpus.

The output from such a neural network would from a given word be the probability that a
word within its vicinity is a certain word (e.g. the probability that dog is the neighboring
word to cat). However, the actual output from the neural network is not what is used
as the word2vec model. Instead, the data produced to predict this probability is what
is being used in the end. The neural network creates data based on a given corpus in
the form of a matrix. What the matrix contains is feature vectors for each word in the
vocabulary. These vectors are then used to create a vector space where words with
similar features end up close to each other. The vector space can then be used to predict
words used in a similar context. It could, for example, be used as a translator between
English and Spanish (given that the input data consists of texts in English and Spanish).
Words that are similarly used in a corpus should then end up next to each other in the
vector space (e.g. one and uno) and can thereby be concluded as to be related within the corpus. The method used to select both names and words to include in the dictionaries has been developed utilizing this model trained on a corpus of texts collected from flashback.se. The method takes a word as input and gives suggestions of words related to it within the corpus it was trained on. Each suggestion given by this model can then be affirmed by the user as related to the given word or discarded. If a suggested word is approved by the user it is added to a list. From this list, the method can then be run again, this time using the list as its basis for predicting related words. In this thesis, the words contained within the lists have already been selected by a group of experts using this method. The names for the list of targets will be selected using this method based on names of individuals that, in earlier studies regarding online hate have shown to be common objects of hate.

### 2.3.2 Prepossessing

In order to get the data working with the methods, as well as creating data suitable for specific purposes, e.g. testing, data needs to be selected and reprocessed. The data used for testing and training the machine learning model consist of texts i.e. posts and comments taken from several different web forums and comment sections. Because of this data being user-generated, in order to use it for training and testing, the comments and posts need to be annotated i.e read and marked as hateful or not hateful. As mentioned earlier this data has already been sourced and used in earlier studies, this part of the process has therefore already been done. The annotation was done by letting psychology students read and mark these text as hate or not hate. This data set consists of 7 290 texts out of which 5 798 contains no hate and 1 492 do. From this, data used for testing in the first stage will be chosen with the criteria that the texts need to contain at least one name from the lists of names produced for this study. Moreover, the entire data set of 7 290 texts, except for those texts used as test data, will be used to train the bag of n-grams based classifier. From the annotated set of texts around 600 texts will be selected on a 50/50 basis i.e 300 texts containing hate and 300 not containing hate, and used as test data. These texts will be selected based on a number of criteria aiming to test different aspects of the classifiers as well as possible on different kind of texts. Because of hate being a subjective term there might be a few texts that have been marked as containing hate that does not and vice versa in the annotation stage. Because of this the texts used for testing will be checked one extra time in order to make sure that there is a consensus regarding the classification made by the annotators.
2.3.3 Ethical handling of data

When dealing with data containing personal information (in this case names and usernames) there is always an ethical aspect to it. The data needs to be handled in such a way that it does not compromise the personal integrity of the individuals who, especially when involuntary, direct or indirect are a part of the study.

The aim of this thesis is not to map behavior nor the opinions of individuals who in one way or another are a part of the material studied. In order to protect personal integrity, the data used in this study has been collected from open sources, sources that does not require a password or a membership of any sort in order to view its content. This also means that the users themselves when publishing a text do so in an environment where they are or should be, fully aware of the public nature of the environment where they choose to express their opinions.

Furthermore, in the texts used as examples in this thesis, no matter if it is made up or a real example from the data, names have as far as possible been replaced with @person# (where # refers to the numbering done if multiple individuals occur in one text). Texts, when directly taken from a real-world source are freely translated from Swedish, sometimes altered in order to preserve content important for the example, which can be said to further add a layer of integrity to the user having written such a comment as it makes it harder to find by, for example, googling it.

3 Related work

As a result of hate in digital environments becoming a rising issue, not only has there been an interest in trying to detect this kind of behavior, but also to study the phenomenon as a whole. This has been done, both within a single subject e.g. psychology [Sul04] and linguistics [EKN18] as well as by using an interdisciplinary approach. There are, furthermore, several different areas of hate that have been focused on e.g. how to detect digital hate [NZHL15, WØVH16, DWMW17, KAP18], on investigating hate and digital hate as a concept [EKN18], on understanding why digital environments have become a media for expressing hate to such an extent that they have [Sul04], as well as on researching how it occurs, and to measure how common it is within a certain contexts [KO18, KAP18]. In psychology there have (as mentioned earlier) been several theories of what hate really is and as a result there are many works attempting to reach a definition of hate [Sel16, RMR05]. The psychological research has to a certain extent been laying the groundwork for how to look at hate, and in extension, how to look for it in a digital context (see section 1.1.1).
When it comes to developing reliable ways for hate detection, several different approaches have been tried in the past. These have mainly, but not exclusively been focusing on hate speech. In [NZHL15] Njagi et al. combines a sentiment dictionary and corpus-based features to develop a rule-based classifier for hate speech detection. Njagi et al. used three different sets of features to recognize hate, trying out different combinations of these features to conclude the most effective method. A broad approach, utilizing all three feature sets in combination with filtering of the corpus to only contain subjective sentences, is shown to be the most effective, reaching a precision around 0.72 and recall at about 0.68. Wester et al. [WØVH16] study the effects of various types of linguistic features for training classifiers to detect threats in a corpus consisting of YouTube comments. In the study the most promising result came from using a combination of several of these features. The study also showed promising results using a bag of words model, that at best resulted in a precision score of 0.7325 and a recall of 0.5943. Davidson et al. [DWMW17] argue that lexical detection methods have low precision due to such classification being based on sentences containing/not containing certain predetermined hate terms. Instead, the use of a multi-class classifier is proposed distinguishing between hate speech, offensive language, and non-hateful texts. Their approach is shown to produce good results at classifying offensive language as well as comments not containing hate, although performing substantially worse when trying to classifying hate speech.

The work preceding this thesis, to a certain extent, includes works written about some of the methods analyzed in this thesis itself, as well as studies where some of these methods have been used. These studies do to a certain extent provide analysis earlier made on the very same methods. These analysis can, and will be used as a basis for comparison of the results produced in this study. There are two papers that mainly fall into this category i.e. [KAP18] and [ISK+18].

In [ISK+18] the method using a lexical sliding window is described and three major errors resulting in false positives are identified. Some suggestions for improvements are also brought forward that later were incorporated into the dependency reasoning method evaluated in this study. Described in [KAP18] is the method utilizing dependency reasoning. Also included in the later paper is a small analysis similar to the one done in this thesis, using a much smaller set of names and no nicknames. Here the method is also compared to the sliding window method. The dependency reasoning method did, in that study, find a lot fewer comments (28) than the sliding window (71) although showing a greater precision than the sliding window. Both methods did, however, when compared to a manual analysis of the same set of texts where 128 hateful texts were identified, find a relatively small percentage of the total number of hateful texts.

It can, furthermore, when it comes to testing and training data be noted that Ross et al. [RMC+16] shows that the agreement among annotators of hate speech seems to be low
and argues that it might not be a simple yes or no question. What they instead suggest is instead a scale upon which can be measured how hateful a message is.

4 Results

The methods were as earlier described tested using two different data sets. The result from the different tests showed a great difference in performance depending on what data set was used. From the result, strengths as well as weaknesses could be seen for each method. The lexical sliding window was excluded from the later test stage and was therefore not tested on the in the wild data. The reason for this was that the lexical sliding window method turned out to perform better in a more desirable way than the two other methods.

4.1 Evaluation of hate detection methods

In the test round the three methods were, as described earlier, run on a test data set consisting of previously annotated texts. This set of text did after selection contains 684 texts. From these texts, a comparison was done between the predictions made by the methods and the true value of a text (i.e the annotated value). From this, the confusion matrix for each method was calculated as well as each of the four measurements. The calculated values can be seen below for each method divided into two tables showing the values of each measure (Figure 14), and the exact outcome of true and false positives/negatives for each method (Figure 15).

<table>
<thead>
<tr>
<th>Measure</th>
<th>Lexical Sliding Window</th>
<th>Dependency Reasoning</th>
<th>bag of n-grams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.69</td>
<td>0.66</td>
<td>0.71</td>
</tr>
<tr>
<td>Precision</td>
<td>0.89</td>
<td>0.96</td>
<td>0.99</td>
</tr>
<tr>
<td>Recall</td>
<td>0.42</td>
<td>0.34</td>
<td>0.43</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.95</td>
<td>0.98</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Figure 14 The collected result of the measurements from all three methods
As can be seen in Figure 14, the dependency reasoning method results in the lowest accuracy and recall, however, the method utilizing the sliding window shows the lowest precision and specificity. That the lexical sliding window shows a lower precision and accuracy than the other methods can (as seen in Table 15) be derived from the fact that the percentage of false positives produced by this method, compared to the other methods, are significantly higher. As discussed in Section 2.1, false positives (rather than false negatives) is what is to be avoided due to a desire to avoid an overestimation of hate in a corpus. In this aspect, the sliding window method therefore performs the worst. It should, however, be taken into account that the number of false negatives produced by this method greatly exceeds the number of false positives. Depending on the area of usage for these methods, this can sometimes mean that the number of false positives is of lesser concern. What furthermore can be derived from this result is that none of the methods, from a corpus consisting of 342 texts containing hate, managed to classify more than close to half of these texts correctly. When detecting hate, these methods manage to overlook a lot of the hateful texts contained within a single corpus. This goes in line with the findings in [PKA18] (see Section 3). The method using dependency reasoning, however, performs better in this study than in the previous one while the lexical sliding window performs slightly worse. The result of the measures also proves to be relatively comparable with the findings made in both [DWMW17] as well as [WØVH16] discussed in related work (see Section 3). The methods proposed in these articles both result in a precision higher than recall. The recall and precision of the methods evaluated in this thesis do, however, in relation to each other differ more, showing a higher precision and a lower recall. What this indicates is that all of the methods used here can be said to be superior at detecting texts that actually are hateful, although worse at detecting hateful texts in general. Moreover, it is interesting to note that the bag of n-grams model used in this thesis greatly outperforms the bag of words model used in [DWMW17] when it comes to precision, but that its recall indicates a 15% worse performance. This indicates that the bag of n-grams used in this study performs worse when it comes to actually detect hateful texts in general but that the texts that it classifies as hateful more often are correctly classified. The method that, no matter what measure is taken into account performs the best at this stage, is the bag of n-grams model.
4.2 Detecting hate in the wild

The data set collected for the methods to be tested in the wild did after collection consists of 7,0806 texts from flashback.se. Because of the lexical sliding window showing a relatively high percentage of false positives, it was at this stage decided not to test it on the larger data set.

In this data set, the bag of n-grams classifier detected 3,3363 hateful texts and 3,7443 non-hateful texts i.e. a ratio of close to 50 percent hateful texts. Considering that the source of these texts is a relatively diverse web forum, especially when it comes to subjects that are discussed, and based on previous studies done regarding hate, it is extremely unlikely that such a large portion of the data set can be classified as hateful. The dependency reasoning method did on the other hand show a result that is equally questionable. With 171 texts classified as hateful and 7,0635 as non-hateful, the hate and non-hate ratio seems, in opposite to the result from the bag of word classifier, as an underestimation.

As explained in section 2.3 a sample of 600 randomly chosen comments from the larger data set was at this stage manually annotated. The sample did after annotation turn out to contain 47 hatefull texts and 553 non-hatefull texts. The annotated values were then compared to the predictions made by the classifiers and the results can be seen in Figures 16 and 17.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Dependency Reasoning</th>
<th>bag of n-grams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0,92</td>
<td>0,51</td>
</tr>
<tr>
<td>Precision</td>
<td>0,66</td>
<td>0,09</td>
</tr>
<tr>
<td>Recall</td>
<td>0,04</td>
<td>0,61</td>
</tr>
<tr>
<td>Specificity</td>
<td>0,99</td>
<td>0,50</td>
</tr>
</tbody>
</table>

**Figure 16** The result of each measurement for the two methods tested in this stage

<table>
<thead>
<tr>
<th>Measure</th>
<th>Dependency Reasoning</th>
<th>bag of n-grams</th>
</tr>
</thead>
<tbody>
<tr>
<td>TruePositives</td>
<td>2</td>
<td>29</td>
</tr>
<tr>
<td>TrueNegatives</td>
<td>552</td>
<td>280</td>
</tr>
<tr>
<td>FalsePositives</td>
<td>1</td>
<td>273</td>
</tr>
<tr>
<td>FalseNegatives</td>
<td>45</td>
<td>18</td>
</tr>
</tbody>
</table>

**Figure 17** The correctness of each classification by the two methods tested in this stage
As can be seen in the Tables 16 and 17, the performance of both methods is relatively poor, especially compared with the results shown in the test phase. Even though the dependency reasoning method shows a quite good result based the on the measures (except recall), it can be said that the measures used here have a potential to show a somewhat biased view of reality. Looking at the true and false positives/negatives produced by both methods in relation to the annotated values, another reality becomes apparent. The number of false positives produced by the bag of n-grams method and the number of false negatives produced by the dependency reasoning method is remarkably high.

5 Discussion

Even though the result from testing the methods in the wild shows such a poor outcome, the comparison between the different tests does open up for some quite interesting examinations of the different methods. It is possible to find a few differences between what kind of texts that each methods finds hateful. The biggest difference in the first stage is between the methods working with a dictionary and the machine learning based bag of n-grams method. As an example of this, the bag of n-grams model finds 67 truly hateful texts that both of the dictionary based methods classify as non-hateful. In comparison, the Lexical sliding window classifies 41 truly hateful texts as hateful that are not classified as hateful by the Dependency reasoning method. What also needs to be taken into consideration here is that there actually is a difference of 30 texts classified as hateful in total between the two dictionary based methods. This can be compared to there only being 2 between the bag of n-grams model and the Lexical sliding window. What this means is that the difference between the dictionary based methods majorly lies in how much they manage to catch (i.e. between false and True Positives), while the difference between the dictionary based methods and the bag of n-grams classifier lies in what texts are classified as containing hate in general. It could therefore very well be derived that these methods manage to find different types of hateful comments. It can also be said that for some of texts classified as hateful by the bag of n-grams classifier but not by the dictionary based methods, there is no apparent explanation to why the dictionary based methods does not classify them as hate. Among these are:

@person är en psykopat som hatar vita svenskar.

Which can be translated to ”@person is a psychopath who hates white swedes”. Although psychopath is in the dictionaries given to each of the dictionary based methods, the name mentioned are part of the name lists, and the opinion is being expressed in a very straightforward and linguistically uncomplicated way none of these methods clas-
5 Discussion

sifies the text given above as hateful. This is most likely an indicator of a minor bug which means that there are at least a slight misrepresentation in the true performance of these methods.

Furthermore, an interesting aspect is that the bag of n-grams classifier when analyzing the test data in the first stage is the method that finds the most threats. Out of eleven comments containing death threats or threats of aggravated assault, the bag of n-grams classifier classifies eight correctly while the lexical sliding window manages to find four and the dependency reasoning method three. The reason for this, when considering how the methods work, is relatively easy to see. Looking at the difference between the texts found by the dictionary based methods, it can be concluded that the issue lies in threats often being expressed in a subtle way. A threat is often expressed using words that are hard to predict and include in a list of predetermined words. This can, for example, be seen in the sentence (classified as hate by the bag of n-grams classifier):

Tänk om man stötte på @person på perongen, det skulle vara så himla lätt att bara ge henne en liten knuff ner på spåret

That translated would mean ”Just imagine meeting @person at the platform, it would be so easy just to give her a little push”, in which there are no apparent words that normally are (or maybe even should) be included in a list of hate words.

The results presented in the previous section shows a difference in performance between the methods when used on the training data and on the larger unannotated data set. It is, however, interesting to compare both of these results. Before such a comparison is made however, some of the aspects from the results shown in the in-the-wild-round needs to be discussed.

The reasons for the poor result in the later part are potentially many. When it comes to the bag of n-grams based classifier, studying its classification of some of the comments can however give some clues. One such sentence is:

Skyl inte @party1s dåliga resultat på @party2. @person måste avgå.

This sentence was by the bag of n-grams method classified as hateful, even though it is not. Translated it means, ”don’t blame @party1’s bad result on @party2. @person must resign”. Breaking this text apart trying to find why it is classified as hateful shows that the bag of n-grams based model does so based on the word ”avgå” (i.e. resign). This conclusion was drawn by letting it classify parts of this sentence alone, removing some parts of it. Always when removing the word ”avgå” this sentence was classified non-hateful.
Even when including avgå in a sentence where it doesn’t make sense, that sentence is classified as hateful, for example:

han hon det jag nej avgå

This sentence is basically made up of disconnected words and would when translated mean ”she he I no resign”. A number of such sentences including and not including the word ”avgå” were tried, even if the sentence could not in any other way be classified as hateful, ”avgå” alone seems to this model to indicate hate. What this most likely indicates is that this word, in the data set used for training, occurs within texts annotated as hateful and that by the tfidf is given a weight that makes it significant in such texts. This turns out to be the case for a large amount of the sentences wrongly classified as hate. When breaking these sentences apart as described above it shows that there seems to be words or phrases that are wrongly indicating hate to this classifier.

This also seems to be the case when it comes to sentences that are actually hateful that have not been classified as such. In these sentences a lack of indication can instead be observed, i.e. that phrases that clearly should indicate hate does not. One example of such a text is

fyfan va rutten @person är

Translated it means ”@person is so fucking rotten”. The phrase fucking could clearly indicate strong emotions and in combination with the word rotten it can be said to be a clear indicator of hate. Despite this, the text is not classified as hate by the bag of n-grams method. This type of sentences, however, are fewer in the sample taken from the last evaluation as there are only 47 sentences that can actually be classified as hate. There are however several examples of this among the classifications made by this method of texts included in the first stage. The most likely reason for some of these texts not being classified as hateful is that the texts in the training data instead, to an not enough extent, contains texts annotated as hateful containing the words and phrases that are overlooked. Although not desirable, an important point to make regarding these errors is that such miss-classification to a certain extent can and should be expected. The large number of these types of errors found in this case, however, indicates that there most likely are some issues that can be dealt with in order to improve the accuracy of the different methods. One likely cause of errors when it comes to the bag of words classifier is that the data used for training, in its current form, is not suited to train a model for hate detection. The set of annotated comments might need to be looked over as it most likely contains too many vaguely hateful or non-hateful texts annotated as hateful. This would lead to the machine learning model interpreting too many non-hateful words as hate indicators and vice versa, as has shown to be the case here. This would go in line with the findings made by Ross et al. in [RMC+16] (see section 3) and has to a certain
extent been observed when examining the texts and their corresponding annotations in the training data. That is also why the test data used in the first stage was manually annotated an extra time before using it. Moreover, there is the issue with hateful texts not being classified as hateful. This could to some extent depend on this issue as well. It is however more likely that there are not enough texts in the training data set containing the kind of words, phrases, or expressions that are overlooked by the machine learning model.

When it comes to the performance of the dependency reasoning method in the later stage, discerning why its performance to such an extent differs from that in the test stage can in most cases be traced to the words contained in the hate dictionary. What the difference of performance between the two phases shows is that the test data does not, in a sufficient way, seem to reflect the diversity of words used to express hate. A direct conclusion that can be drawn based on this is that the same goes for the dictionaries used. An important point to make however, is that including a large diversity of words that potentially could indicate hate also increases the risk of creating more false positives. When including words that just as well as expressing hate could be used to express something completely non-hateful, there is a risk of classifying non-hateful texts, containing these words, as hateful.

Furthermore, although the Lexical sliding window was not used on the larger data set it is, due to the dictionaries used by both the dictionary based methods being almost identical, very likely that its performance would have been similar to that of the dependency reasoning method. Although considering the performance of this method in the test stage it is likely that it would have been able to detect a few more hateful texts but also producing a larger number of false positives.

Despite the fact that many errors can be traced to the data used to build the hate detection methods, some sources of incorrect classification do not lie within the data, but the methods themselves. An example of this is the large number of false positives produced by the lexical sliding window method. This type of errors produced by this method can mostly be derived from the non-contextually way this method is basing its assessment on. If a word from the dictionary and a name on the lists appear within in a sentence, this method does not take into account whether the word is actually directed at the target or not nor if it is even used as a hate word at all. An example where that is the case is the sentence:

**Och så fick mamma läsa på den där texten att @person var mordad.**

Which can be translated in to "And so my mother got to read the text about @person being murdered".
The lexical sliding window classifies this sentence as hateful due to the word murdered even though it is not used in a hateful way. As the semantic role of words are being ignored these errors are prone to appear in classifications made by this method. The same goes for negations (e.g. I hope @person doesn’t gets murdered) and quotes (e.g. @person claimed he was raped). These types of false positives are not a problem when it comes to the other methods. This especially is the case for the dependency reasoning method as it, to a much larger extent than any of the other methods, takes into account the semantic relations between words and targets.

5.1 Ethical aspects

As mentioned in Section 2.3.3 (Ethical handling of data) ethics is, when dealing with these types of methods, an important aspect to take into consideration. Although the handling of the actual data in relation to this thesis has been discussed, there is a much larger context in which these methods (as well as similar methods) and their purpose could be questioned from an ethical perspective.

A subject for discussion when it comes to ethics today is privacy in relation to surveillance. In many cases, this discussion takes its basis in surveillance for a greater good versus preserving our right to maintain our personal privacy. The methods discussed in this thesis could very well be used in such a way that intrudes on personal privacy, for example to monitor user behavior. Such monitoring can without to much strain of thought be seen as a potential form of surveillance. Applying these methods actively, for example, as soon as a comment or post is made in web forum or comment section to monitor hateful content is as per the definition of the word, a form of surveillance. In the context of surveillance and the potential violation of personal integrity, what needs to be considered is how to judge when, or if at all, the use of such methods can be justified. Marx does in [Mar98] argue that the ethics of this kind of activity should be judged according to certain criteria. Marx argues that the means and conditions of data collection, how the data is used and what goal the user hopes to achieve needs to be taken into account, but that personal integrity as well as dignity always needs to be considered in relation to this. As mentioned previously in this thesis the problem of hateful and threatful content is of importance, not only on an individual level but also in a democratic context. Arguing with the basis that the end justifies the means, developing methods to monitor hate could very well be justified. What needs to be considered, however, is where to draw the line. There are today real examples where monitoring unwanted behavior is used to control individuals on a much larger scale than just their activities in for example an online forum. An example of this is in the social credit system that is right now being rolled out live by the Chinese government [Ram18]. In this system, the behavior of each citizen in a number of contexts (including online activities) is being reflected on a
societal level by a rating system. In the system, desirable behavior is credited and undesir- 
sirable behavior penalized by lowering an individuals rating in the system. The rating 
then affects an individuals possibilities within society, keeping individuals with a low 
social rating from benefits and rewarding those with a high social rating (e.g by making 
it easier to apply for an apartment or a job). The system itself is a way to rate the trust of 
citizens and to encourage behavior desirable within a well-working society. It could, as 
an effect of this be said that the system is used for the greater good. However, the system 
can very well be argued to undermine the personal integrity to an unacceptable extent, 
even if so in the interest of an entire nation. Although an extreme example, it is not far 
fetched to think that tools like the ones presented in this thesis could be used in such a 
system. It is in such a situation that one could argue that a line has been crossed. How-
ever as earlier discussed, as much as the monitoring and restriction of hate speech could 
be seen as a means of restricting the freedom of speech, there are many ways in which it 
can be seen as a way of preserving it. When hateful speech and threats become a reason 
for individuals not wanting or feeling able to express their opinion, it becomes a threat 
against the very foundation of a democratic society. Furthermore on a level of national 
security, monitoring hateful behavior directed against certain exposed individuals (e.g. 
the prime minister) could potentially be used as a method of risk assessment.

What this goes to show is that determining the ethics of using such methods as the ones 
presented in this thesis is not an easy task. The most sensible way of determining this is 
probably to, depending on the situation, try assessing the ethics of the situation itself. It 
should, however, be said that the development of such methods as the ones presented in 
this thesis to a certain extent facilitates the use of such methods for purposes ethically 
questionable.

6 Conclusion

In this thesis the performance of three methods for hate detection have been evaluated. 
What the results show is how tricky the development and application of such methods 
can be. Hate is a broad concept that is extremely hard to define in general and even 
harder to define in such a way that an algorithm can be designed or taught to detect it. 
Each approach presented in this thesis comes with various disadvantages and problems. 
When using a dictionary-based method the main problem is what words to include in 
the hate dictionaries. Covering the extent of how hate can be expressed by listing hateful 
words and phrases is foremost a matter of finding the right words. What this means is 
that the dictionaries need to be limited in such a way that the false positives are kept to 
a minimum while still covering hateful words and phrases as much as possible.
This is somewhat paradoxical, and to develop a dictionary with enough coverage to give a realistic estimation of hateful content in a corpus is a task that comes with many challenges.

As can be seen by the result produced in this thesis, machine learning also has its disadvantages. A common saying when it comes to machine learning is that the model is only as good as the data it is fed. That hate is such a subjective term does in this case mean that good data is a somewhat subjective term. When extracting data used to train a model to detect hate, what needs to be considered is where to draw the line for what should be classified as hateful and to work by that definition. This also is per the nature of hate a very challenging task. As discussed in the last section the quality of the training data is to a large extent related to how correctly it has been annotated. The agreement of annotators could, however, differ quite a lot. Despite having a clear definition of hate it could very well be so that one annotator feels that just mentioning the word homo is an indicator of hate towards homosexuals, while another annotator does not.

What can be concluded is that the problem of the methods does not seem to lie within the methods themselves as much as it in the data provided. When it comes to the dictionary based approaches it is mostly, although not exclusively, the dictionaries that seem to prevent the detection of certain hateful content. For the machine learning based bag of n-grams, the method seems largely to fail because of the shortcomings of the training data which mostly depends on the quality of the annotation.

On a more positive note, it can be concluded that the use of dictionary-based methods in combination with machine learning, have the potential to detect a broader spectrum than what could be done using them separately. Machine learning has the potential to detect texts containing expressions that, although common, is hard to include in a dictionary. The dictionary-based methods are in turn better for detecting texts containing words or phrases that are not commonly used but nonetheless hateful.

7 Future work

As mentioned in the conclusion a combination of the methods presented in this thesis could very well be a viable way of further development. Considering the low number of false positives produced by the dependency reasoning method, the bag of n-grams classifier in combination with that method has the potential of catching a much wider range of hateful texts than those methods could by themselves. The methods and data used do however require some fine-tuning to give a reasonably realistic view of hateful content in a given corpus.
The most important aspect when it comes to improving these methods seems to be the data that the methods need to operate, i.e. training data and the hate dictionary. For hate dictionaries to work it might be a good idea to expand the dictionaries. For this to be a viable solution, more empiric studies might need to be done regarding the actual usages of certain words. To test the improvements, the test data also might need to be considered. What was shown in this thesis, is that the data chosen to test the methods in the first stage showed a false representation of the performance of each method, at least in relation to the data set used in the later stage. The most likely cause of this, as earlier discussed, is that the test data to a not enough extent reflects the wide spectrum of how hateful texts actually looks in the wild.

For the machine learning method, the main fault when it comes to the training data seems to be the annotation of comments. When annotating comments it could, therefore, as a simple solution be a good idea to show a few examples to the annotators of comments belonging to the different classes (i.e. in this case hate/not hate). It could furthermore be a good idea to hold a class with the annotators that, in relation to a set definition explains what is to be sought for in the texts given to them. Another approach is to only use data annotated by experts or data that has been read through by experts i.e. annotated by students and double checked by experts.

Another solution could be to try other approaches to hate detection than the ones presented in this thesis. There might be other algorithmic approaches that work better than the ones used by any of the dictionary-based approaches. It could, for example, be so that there are other linguistic methods working better than the one used by the dependency reasoning method. Furthermore, despite having tested different machine learning models there might be other models that work better than logistic regression. There could also be better ways than the bag of n-grams method coupled with a tf-idf for producing the raw data put into the model for training. As this method does not take into account the semantic roles of the words there might be other ways of producing raw data that does. This could potentially help the machine learning model to further recognize when hateful words are not used in a hateful context. There might also be ways of tweaking the existing methods e.g. adding or chaining a layer of reasoning to the dependency reasoning method or to change some of the parameters of the bag of n-grams based method.
References


References


Natural Language Processing for Computer-Mediated Communication (NLP4CMC), 2016.


