Identifying regrettable messages on social networks

Patelski, S.P.G.

Award date:
2015

Eindhoven University of Technology

Department of Mathematics and Computer Science

Master's Thesis

Identifying Regrettable Messages on Social Networks

Author:
Stefan Patelski

Supervisors:
dr. Mykola Pechenizkiy (Eindhoven University of Technology)
dr. Aristides Gionis (Aalto University)

Assessment Committee:
dr. Mykola Pechenizkiy
dr. George Fletcher
dr. Dragan Bosnacki

August 31, 2015

Abstract

The things you post on the internet can stay there for years, or maybe even forever. People often forget that their posts on online social networks can be read by more people than they think. Even things that are posted privately to a group of friends may leak to a larger audience when the post is spread further by one of the friends. According to one study, 41% of Twitter users regret placing certain tweets. When such a tweet is posted, only 11% realizes immediately after posting that it was a mistake. This may cause harm to their careers or relationships. Therefore it is reasonable to say that social network users could use a little help with determining what to show on the social network. Our study makes a first step towards finding a methodology that enables users to detect regrettable messages on social networks. We imagine that this methodology might be used in the future by products that either warn a user before posting, or that remove sensitive posts from a social network account in bulk. We have implemented a proof of concept using a small data set, gathered through crowdsourcing. We use classification techniques from machine learning to classify tweets as sensitive or not sensitive. The classifier in our implementation is able to correctly classify tweets in 57% of the cases. This is not yet a usable classifier, but it illustrates that there is a potential to solve the problem. Using better data and better feature extraction, a good solution could be developed based on our proof of concept.

Contents

Chapter 1

Introduction

We started this study with the idea to improve privacy with the help of techniques from data science. We narrowed this down to detecting regrettable messages on social networks. In this introduction we will introduce the reader to the subject and explain our motivation.

1.1 What is Privacy?

According to Malhotra et al. [11], there are three dimensions for the privacy concern of internet users: collection (of personal data), control (e.g. have the choice to opt-out), and awareness (of privacy practices). Based on this we can say the following: Internet Privacy means to be able to know and determine whether, to whom, and for what purpose your personal information is used. However, Acquisti and Grossklags [4] found out that people have little knowledge about the positive and negative consequences of disclosure of personal information. Furthermore, people do not usually exhibit rational economic behavior when it comes to information disclosure [5]. From this we can conclude that internet users may need some help to protect their privacy. They might place a message on a social network that they later regret sharing. So for the purpose of this study, we have focused on public statements that might cause the author to regret having made this statement.

1.1.1 Regrets on Online Social Networks

To provide further indications that internet users may need some help to protect their privacy, we will discuss some research about regrets on social networks. Wang et al. conducted a survey about regrets regarding posting to Facebook [20]. They summarized [18] the study as follows:

> *We see that regrettable postings revolve around sensitive topics (e.g., alcohol consumption, sex, politics, religion) and content with strong sentiment (e.g., arguments and criticism).*

Sleeper et al. [15] surveyed regrets that people have about posting Twitter messages (tweets). Of the participants in the survey, 41% said they regretted placing certain tweets. The survey found the following types of regret to occur most often:

- Reveal too much (25% of all regretted posts)
- Direct criticism of a person (20%)
- Expressions of feeling and emotion (14%)
- Direct attack (13%)
- Blunder (11%)

They also found that in 58% of the cases, the participant realized the reason for regret by themselves. In the other cases, the behavior of someone else makes them realize that there is a reason to regret placing the Tweet. In the cases that they did realize the mistake themselves, they usually didn't realize it immediately. Only 11% realized it immediately after posting. This indicates that in some cases it is necessary to alert users about possible future regret, because they are not directly able to realize it themselves. The authors point to earlier research which visualized sentiment in email messages, to make the writer aware of how others might perceive the message that he is writing [9].

1.2 Potential Applications

As has been discussed in Section 1.1.1, many people come to later regret some message that they post on a social network. One way to deal with this regret is to prevent the author from publishing the message in the first place. This would require a system that analyzes the message before it gets posted online, and warns the author in case the message is predicted to be sensitive. Another way to deal with the issue is to remove social network posts after they have already been published. Some or many people may have already seen the message, but there might be scenario's in which one would want to remove sensitive messages in bulk. For example, it is common nowadays that employers look at the social network pages of job applicants to find information about the candidate. It would be useful for the candidate to clean up their social network accounts. A system that automatically detects potentially sensitive messages could greatly reduce the amount of work required to do this, especially when the job applicant is a very active user.

1.3 Problem Formulation

The role of this study is to make a first step towards finding a methodology that enables us to answer the following question:

Given a message on a social network, is this message regrettable or not?

Messages can be shared to the entire world, like they usually are on Twitter, or with a large group of Facebook friends. When we say that some message is *regrettable*, we mean that it is a message that the author might better not make (or keep) public. The reasons for these can be anything that could potentially cause harm to the author. The message might affect relationships or it might result in stolen identity. See Figure 1.1 for a visualization of the problem.

Because our problem is to find which of two classes a piece of text belongs to, we treat this problem as classification task. We identify some of the features that can play a role in predicting the right class, and we provide a proof of concept.

Throughout this report we use the words *regrettable* and *(privacy) sensitive* as synonyms.

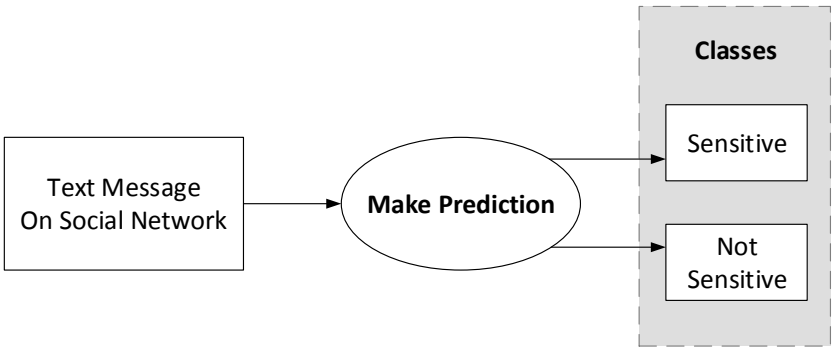


Figure 1.1: A visualization of the problem.

1.4 Summary of Methodology and Results

We try to make a prediction of the regrettability of a message on a social network by extracting features to be used by a classification algorithm. These features can be ready-to-use numerical properties like the time of day. But some features can be extracted from the message text using more complex methods. This set of complex features includes sentiment score, language quality, and controversy score. To find training data for our classification algorithm we let paid workers on a crowdsourcing platform label Twitter messages. Our results show that both sentiment score and language quality help with making a prediction. Our current implementation of the controversy score however was of very little use to the classification. Overall we see that the idea is promising, but needs more training data. We also suggest investigating the potential of other features.

1.5 Thesis Structure

In Chapter 2 we formulate the problem and propose an approach to solve it. In Chapter 3 we go into detail about a the controversy score, which is a feature that is not trivial to acquire. Then in Chapter 4 we explain how we gathered the data to give a proof of concept of the approach. In Chapter 5 we analyze the gathered data to show how well the approach performs and which factors contribute to its performance. We finish by reviewing related work in Chapter 6 and by providing a conclusion and suggestions for future work in Chapter 7.

Chapter 2

Problem Formulation and Approach

In this chapter we explain the problem, and our plan to solve it.

2.1 Problem

Our problem is to find the regrettability of a given message on a social network. We assume that we have access to all information about the message. Not just the text, but also information about the author's user account and meta-data about the message, such as the amount of times it was reshared by others and the time that it was published. We treat this problem as a classification problem, and expect that this will enable us to solve the problem. To make it a bit easier for us, we don't consider the content of hyperlinks in the message. Nor do we look at messages that contain pictures or videos, as this would make the project too broad.

2.2 Scope

The theoretical implications of this study should be able to be generalized to all social networks with messages that contain short text. However, we have chosen to narrow our scope to just Twitter. This way we only have to deal with the API (application programming interface) of one social network. We have chosen Twitter because messages are public by default, whereas other social networks are often private by default. Twitter also has a very extensive and well-supported API.

2.3 What Do We Need?

A trivial approach to alerting users of potential privacy danger would be to raise a flag whenever one of the following contents is detected in the message:

- A controversial topic
- Strong sentiment
- Bad language quality
- Sensitive personal information

With this approach we are required to make the assumption that each of the above features is equally harmful to the user's privacy. It also ignores potential cues from features that are more ambiguous in their predictive power. Examples of this include the presence of the following:

- @-Mentions
- Hyperlinks
- Timestamp
- Retweets
- Length
- Hashtags

In order to find out which features people find to be harmful to their privacy, we could hold a questionnaire. The downside of this is that it is difficult to include all possible features in the questionnaire. E.g. it might be the case that the timestamp of a tweet is a very good predictor of its sensitivity, but we won't find this in a questionnaire.

2.4 Proposed Approach

Please have a look at Figure 2.1 to see a schematic overview of our approach. Our starting point is a Twitter message (tweet). The goal is to classify this tweet as either sensitive or not sensitive. We do this by extracting features from the tweet. These features should correlate with the outcome so that they can be used by a classification algorithm. This algorithm is fed with training data that we have gathered through a survey.

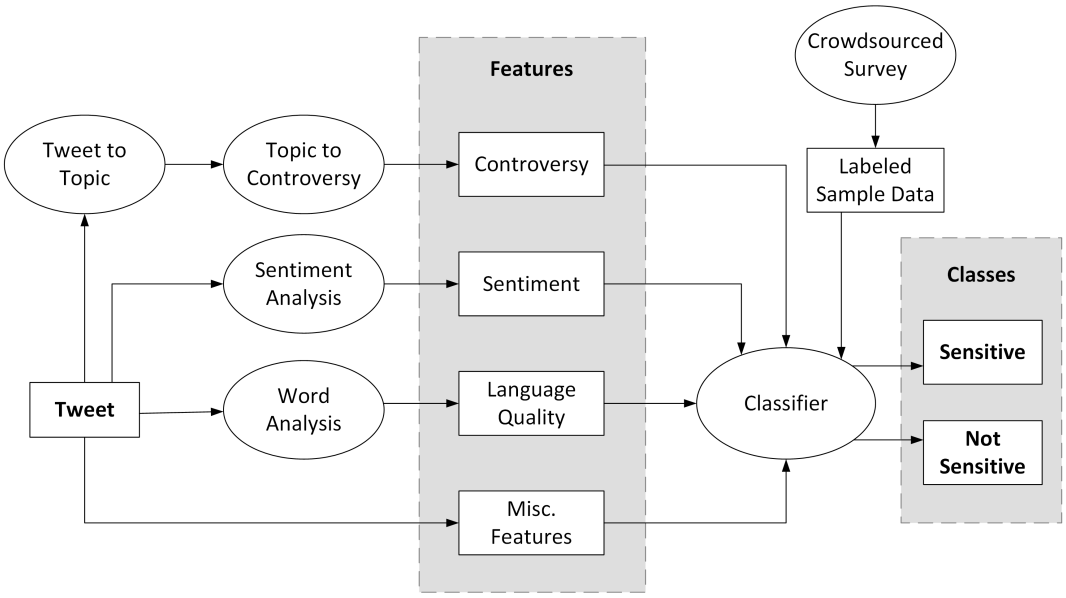


Figure 2.1: Architecture of our approach.

2.5 Features

The following subsections describe the main features that are used as possible predictors for sensitivity. Some of these features are inspired by the features used in [6].

2.5.1 Controversy

We use the method of Sumi et al. [17] to find the controversiality of a certain Wikipedia page, which can be used as a proxy for tweets that talk about the topic of this Wikipedia page. Their method is based on the amount of mutual reverts, which means that user A makes an edit to a page which is then reverted by user B, and user B makes an edit to the (same) page which is then reverted by user A. The mutual reverts are then weighted by the experience of the least experienced user of the pair. Experience is defined as the amount of edits an editor has made. This is done in order to prevent vandalism to affect the controversiality measure.

In order to make this feature work, there needs to be a good mapping of tweets to Wikipedia topics. We have considered several methods for this:

- Counting the occurrence of tweet-words in the wiki-text [8].
- Using the Yahoo content analysis API, which returns relevant Wikipedia pages.
- Using search engine techniques.
- Named entity recognition (possibly as support for the other methods).

The first three of the above methods were tested. We settled on using a Lucene-based search engine. The topic matching of the search engine approach did not have the highest accuracy, but it was the fastest method. More details about this can be found in Section 3.

2.5.2 Sentiment

We use a very simple SentiWordNet-based sentiment analysis to find the neutrality of a tweet [7]. That is, how strong an opinion is expressed in a tweet, ranging from neutral to highly opinionated. Our implementation retrieves the subjectivity score per word from a publicly available list of SentiWordNet-score. We then take the average score over all words in the text that are not stop-words.

2.5.3 Language Quality

Duan et al. [6] have described a language quality feature based on the relative amount of dictionary words in a tweet. We have made an implementation of this.

2.5.4 Miscellaneous Features

These features are included as meta-data provided by the Twitter API.

- Length
- Timestamp
- Number of @-Mentions
- Number of URLs
- Number of retweets
- Number of hashtags

- Number of financial symbols
- Number of times favorited
- Number of previous tweets by user
- Number of followers
- Number of people followed
- Follow ratio
- Whether the tweet is a reply
- Whether the tweet contains a location

2.5.5 Sensitive Information

Features that are not implemented but could be found in the tweet-text:

- Gender
- Age
- Education
- Income
- Date of birth

2.6 Classification Algorithm

The leading algorithms for classification with a training set of smaller than 100000 samples are Naive Bayes, Support Vector Machines, Decision Tree Learning, and Nearest Neighbors. We have implemented these classification algorithms from the Scikit-learn Python library [14] to test which works best. These algorithms were used to train a model using data gathered through a user experiment.

Chapter 3

Controversy Score

In Section 1.1.1 we discussed research that showed that certain sensitive topics, such as politics, religion, or sex, are among the main causes of regret on social networks. So if we can automatically detect these sensitive topics in messages on social networks, we might be able to better predict that an author may regret placing the message.

In this chapter we discuss our implementation for finding a controversy score, that indicates how controversial a tweet is. This process requires two steps. First the tweet needs to be mapped to a topic. After that the topic needs to be mapped to a controversy score. We discuss evaluations of both steps. In Section 5.3 we evaluate how well the controversy score predicts the perceived sensitivity of the tweet itself.

3.1 Mapping Tweets to Topics

To get the controversy score of a tweet, we first need to map the the tweet to a topic, which can later be mapped to a controversy score. Mapping a tweet to a topic is difficult, because there is only a small piece of text to work with. We have considered three methods, namely the following:

- Counting the occurrence of tweet-words in the wiki-text [8].
- Using search engine techniques.
- Using the Yahoo content analysis API, which returns relevant Wikipedia pages. [2]

We have implemented all three of these methods to see how well they work. We will describe the implementations of these three methods in the sections below.

3.1.1 Tweet-Word Occurrence

The tweet-word occurrence method works by simply counting how often the words from the tweet (excluding stop-words) occur in the Wikipedia text, normalized for page length. The Wikipedia page that has the highest count of tweet-words is considered to be the best match. To speed up the process, we only consider a set of candidate pages. These candidates are the pages corresponding to all words in the tweet. So if one of the words in the tweet is "Eindhoven", then the Wikipedia topic about Eindhoven becomes a candidate. When a word has multiple possible meanings, Wikipedia usually serves a disambiguation page that display

multiple options. The name "Springfield" for example can refer to many different cities. In this case we add all possible pages to the list of candidates.

This method is a part of a technique by Genc et al. [8]. Their technique involves first using the tweet-word occurrence method which is described in the previous paragraph. This Wikipedia topic, which can be any topic on Wikipedia, is then matched to another Wikipedia topic from a short predefined list of topics. For example, we might find that a tweet is on the topic "Barack Obama", and our list of possible topics might be "Politics", "Religion", and "Sports". If the technique works correctly then it should match "Barack Obama" to "Politics". Thus the tweet is matched as "Politics". The matching from one Wikipedia topic to another was done by Genc et al. by looking at the categories of the topics. These categories are structured in a graph. The distance function between two topics can thus be defined as the length of the shortest path in this graph between the topics. Genc et al. evaluated this technique with tweets that could be classified in one of three topics. The topic "J.D. Salinger" was classified correctly in 93% of the cases, "iPad" in 87%, and "Haiti" in 80%.

3.1.2 Yahoo API

The Yahoo content analysis API can be accessed by HTTP request. This request should contain a query in the Yahoo Query Language, which looks as follows:

```
select *
from contentanalysis.analyze
where text="tweet text goes here"
```

This query returns the content analysis in JSON format. We use a standard JSON parser to retrieve what Yahoo thinks is the most relevant Wikipedia topic for each tweet. The Yahoo API might be a good way to find the Wikipedia topic. But before we could experimentally test the performance of the three methods, it turned out that the structure of the returned output of the Yahoo API had changed. Because of that our implementation did not function any more. This illustrates the downside of relying on a closed API. For this reason, and also because of time constraint, we have chosen not to research this option any further.

3.1.3 Search Engine

We implemented a local offline search engine using Apache Solr [1], a platform based on the Lucene search engine. The search engine index contains the plain text versions of the English version of Wikipedia. It was indexed once using an xml dump file from Wikipedia. The index is stored on an external hard drive attached to the computer that runs the search engine. The entire tweet text is sent as a query to the search engine, but first it is filtered from the following:

- non-alphanumeric characters
- AND, NOT, and OR (search operators)

And we added the following search filters to the query to exclude pages that are not useful:

- -"may refer to"
- -"can refer to"
- !title:Category*
- !title:Portal*
- !title:Template*
- !title:File*

- !title:Wikipedia*
- !title:Module*
- !title:Book*
- !title:List*
- !title:*disambiguation*

The search engine returns a list of results, of which we use the first three. That is, we take the first three predicted topics, and calculate the controversy scores. The average of these controversy scores is used as the controversy score of the tweet. This makes the score less precise, but increases the recall of controversial topics.

3.1.4 Evaluation

To evaluate the methods described in this section, we manually looked at the results for 50 tweets. We graded each result on a scale from one to ten, using the following guidelines:

- 10: The result perfectly matches the main topic of the tweet
- 8: The result is a topic that is very similar to the main topic of the tweet
- 6: The result is a side-topic or a topic that is related to the main topic of the tweet
- 4: The result is slightly related to the main topic of the tweet
- 2: The result belongs in the same content category as a topic of the tweet

It should be noted that the evaluated tweets are randomly selected English-language tweets. A lot of them don't have a clearly definable topic, or are about a topic that is very unlikely to have a Wikipedia page. We chose to leave these tweets in the evaluation data because they are an accurate reflection of reality.

In table 3.1 we show the results of the evaluation. We show the grades for each individual result of the search engine. But because we use an aggregate of the three scores to calculate the controversy score (see Subsection 3.1.3), we also included in the table the average grade of the best results (out of the three). This makes the comparison somewhat unfair, but it gives an indication of the ability to have a high recall of controversial topics. As can be seen in the table, the tweet-word occurrence method is the most accurate, but it is a lot slower then the search engine. Section 3.1.5 gives some suggestions on how to make this method faster.

Method	Average Grade	Average Time to Compute (s)
Search Engine (1st result)	1.92	5.18
Search Engine (2nd result)	1.90	5.18
Search Engine (3d result)	1.70	5.18
Search Engine (best result)	2.58	5.18
Tweet-Word Occurrence	2.96	16.46

Table 3.1: The accuracy (grade) and computation time (seconds) for each mapping method

3.1.5 Future Research

The search engine implementation could be improved. The offline search engine implementation that we made turned out to be slower than expected, so instead it is possible to use the API of a commercial online search engine like Google or Yahoo. These search engines have much more resources and data available, and can therefore give better results.

The tweet-word occurrence method works better than the search engine, but it is also slower. The reason for this is mostly because all candidate pages need to be downloaded. So the method can be sped up by filtering the candidate pages, or by making the pages available offline in a fast database.

We have implemented only the method in the first paragraph of Section 3.1.1, not the entire technique that Genc et al. described. We did this because the entire technique maps tweets to a short predefined list of topics, while our implementation maps tweets to any Wikipedia topic. The latter is in theory more precise for the purpose of eventually finding a controversy score, but in the evaluation we have seen that it is also very prone to errors. We should consider that a more high-level topic such as "Politics" might be good enough to get an indication of how controversial the tweet might be. So it might be worthwhile to implement the entire technique described in section 3.1.1, use it to get the controversy scores for tweets, and see how this compares with what we did.

Other than the three methods that we described in this section, it might also be interesting to look at research concerning Named Entity Recognition(NER), possibly as a support for the other methods. Research by Patra et al. [13] shows how NER can be applied to tweets.

3.2 Mapping Topics to Controversy Score

Sumi et al. [17] created a method of finding the controversiality of a Wikipedia topic. Their method is based on the amount of mutual reverts, which means that user A makes an edit to a page which is then reverted by user B, and user B makes an edit to the (same) page which is then reverted by user A. The mutual reverts are then weighted by the experience of the least experienced user of the pair. Experience is defined as the amount of edits an editor has made. This is done in order to prevent vandalism to affect the controversiality measure. Sumi et al. only validated their method on 40 Wikipedia pages, and their method might be not as accurate on lesser known subjects. For our research we have used the scores generated by Sumi et al.

Chapter 4

Survey

In order to provide a proof of concept we have gathered a small data set using a survey that was conducted on a crowdsourcing platform.

4.1 Introduction

We want to find out which features of a tweet predict the regrettability of that tweet. Therefore we need a set of tweets, with enough variance of feature values, to be scored by users based on their regrettability. The features are e.g. controversiality, sentiment, language quality, and miscellaneous such as time-stamp, mentions, etc. The set of tweets also needs to have enough samples of both regrettable and not regrettable tweets.

To make sure that e.g. correlation between time-stamp and sensitivity is accurately captured, we need real tweets. It is not a good idea to use fabricated tweets to specifically test certain features, because these fabricated tweets might not contain the same correlations between features and regrettability as in real tweets.

4.2 Hypotheses

We know from literature that controversial topics can be a cause of regret about posting a message on a social network, so if our controversy score works as intended, we should see that the following hypothesis is true:
1) Tweets that have a high controversy score, will receive a higher sensitivity score from the human labelers.

We also know that expressions of feeling and emotion, and attacks on other persons are causes to regret tweeting. Both these things are likely captured by the sentiment score. So we expect the following:
2) Tweets that have a high sentiment score, will receive a higher sensitivity score from the human labelers.

The high popularity of celebrity magazines and the high amount of money being paid for *scoops* about the private lives of famous people are an indication of how fascinated humans are about learning private information about others. So we expect private information to be a popular category of retweets. This is formalized in this hypothesis:
3) Tweets with a large amount of retweets, will receive a higher sensitivity score from the human labelers.

Reasons for regret also include direct attacks and criticism of a person. When addressing another person on Twitter, people usually use an @-mention. This is a way to alert the other person of the fact that they are being mentioned in a tweet. We think therefore that the following might be true:
4) Tweets that have an @-mention, will receive a higher sensitivity score from the human labelers.

Each individual thinks different about what is sensitive and what is not. But there are specific reasons for people to think that a tweet is sensitive. If none of these reasons are applicable to a tweet, than most people will probably agree that it is not sensitive. But if there is a potential reason for possible regret, then it depends on the person whether he thinks this potential reason is a real reason for him to regret the tweet. So we suggest the following:
5) Tweets that are on average considered more sensitive, have a higher variance in their rated sensitivity.

Privacy sensitivity is subjective, and also dependent on the preference of each individual. Therefore we would like to test the following hypothesis:
6) To predict the sensitivity of tweets from one individual, a model trained with data from the individual will perform better than a model trained with an equally large subset of data from the global data set.

4.3 Data Collection

The most important goal of the survey is to gather representative labeled tweets. This collection of labeled tweets will provide us with the data we need to provide support or help reject out hypotheses. Each tweet consists of text, meta-data, and a sensitivity rating. The sensitivity rating is a value between 1 and 7, representing either not sensitive or very sensitive. In the survey, this data can be gathered by showing participants a subset of tweets, one by one. For each tweet the participant should indicate sensitivity on a seven-point scale, ranging from "not sensitive" to "very sensitive". We aim to gather a data set consisting of real tweets. These tweets can be in one of two classes: regrettable, or not regrettable. We first gather a set of tweets that are representative examples of the respective classes that they should belong to. These tweets need to be labeled as sensitive or not sensitive. This will be done by human labelers employed via Crowdflower, a service similar to Amazon Mechanical Turk, but with much more quality control settings. We ask each participant the question "*Indicate how much you agree with the statement that the author might regret posting this message publicly on Twitter*". They can give their agreement on a seven point scale. We can gather about 3000 judgments for 32 dollars. The human labelers are subjected to test questions of which they have to answer over 75% according to our expectations. This ensures that we filter out labelers who don't take the work seriously. As our final goal is to have two classes of regrettability, there needs to be a cut-off point in the seven-point scale. We discuss this cut-off in Section 5.1.

4.3.1 Training Data

To indicate how well our classifier performs, we use the precision and recall metrics. If 100 tweets are predicted to be sensitive, then the precision indicates how many of those tweets are correctly predicted sensitive. On the other hand, if there are 100 tweets in our validation data that are sensitive, then the recall indicates the percentage of those 100 tweets that are

predicted to be sensitive. The goal of the study is to warn people against sensitive messages, so we want to have a high precision and recall of sensitive messages. To get the right balance between precision and recall it is good to have a balanced training set. That is, a data set that has a 50/50 class distribution. To get such a balanced labeled data set, we can throw away data points from the class that has the most data points, until we reach exactly the same amount of data points per class. But there are unnecessary labeling costs involved when we have to throw away too many data points. Therefore it would be useful to first get an unlabeled data set that is equally balanced. It is difficult to say beforehand whether it will be equally balanced after human labeling, but we can make educated guesses about this.

4.3.2 Validation Data

Our validation data should be a representative sample of reality. Using cross-validation we can use validation data that functions as training data at the same time.

4.4 Preliminary Studies

4.4.1 Survey Design Test

The survey design was tested in iterations, where in each iteration we let about 100 tweets get labeled. During these tests we found out that asking whether a tweet is privacy sensitive does not work well, because privacy can be defined in many ways. Therefore we chose to ask instead how likely it is that the author of a tweet might regret placing the tweet.

4.4.2 Sampling Approach Test

In order to get a balanced data set with an equal amount of sensitive and unsensitive tweets, we should make a pre-prediction of how the tweets will be labeled. The difficulty of this is that predicting which tweets are sensitive is the original problem that we began with. However, our pre-predication does not need to be as accurate as what we eventually aim for. Thus we tried the following. Of a large collection of tweets, we calculated sentiment and controversy scores. Tweets with a high controversy or sentiment score are pre-predicted to be regrettable. Let's say this class contains x tweets. We then selected an x amount of the remaining tweets to be in our other pre-predicted class. We tested this approach in a preliminary study, by letting human labelers classify 200 tweets. If this sampling approach works then we should see a positive correlation between controversy score and sensitivity rating. Instead the correlation value is -0.02, so there is hardly any correlation. For the sentiment score the correlation is only 0.07. Therefore we have decided to not use this sampling approach for the full-scale survey.

4.5 Final Survey Design

We used crowdsourcing platform Crowdflower to let paid workers judge a total of 6000 tweets. 1000 of these were judged during a trial run which had the same parameters as the final run, with the single difference being that they have only three judgments per tweet instead of five. How the survey looks like to the workers can be seen in Figure 4.1.

A difficulty is that participants can only judge the tweet itself, without being aware of earlier tweets of the user. Another problem is that participants have not written the tweet themselves, so they have to try to imagine how the author might feel.

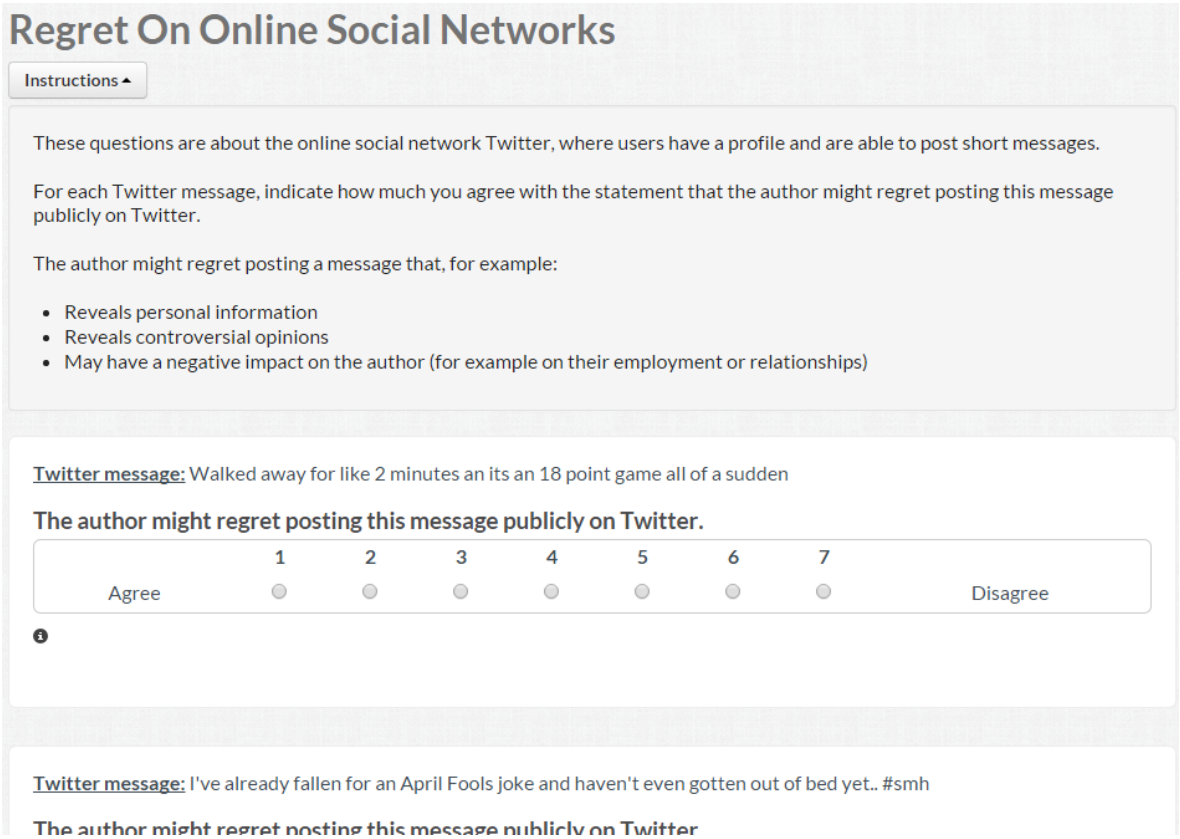


Figure 4.1: How our survey looks like to workers on the Crowdflower platform.

Chapter 5

Data Analysis

In this chapter we analyze the data gathered in the final experiment. This data set includes 28272 judgments about 6000 tweets. In this chapter, we provide an overview with general statistics, before taking a deeper look at how useful each part of the data is.

5.1 Discretization

The sensitivity judgment from our workers on Crowdflower is a number from 1 to 7. This number indicates how certain they are that a tweet might be regretted. We interpret this as a proxy to a scale from low to high sensitivity. But our model uses binary classification. So we need to discretize the rating from a scale of 1-7 to a scale of 0-1. The mean rating is 2.38, and the median is 2. So we considered cut-off points around this value, while taking into consideration that the discretization should also stay true to the real meaning of the feature. That is, if the discretized rating has a value of 1, it should mean that this is indeed a sensitive tweet.

In figures 5.8 and 5.9 we have on the left hand side of the page a discretization that defines all ratings from 4-7 to be sensitive, while the discretization used on the right hand side defines ratings from 3-7 to be sensitive.

5.2 General Statistics

The distribution of judgments is highly skewed towards non-sensitive. This can be seen in Figure 5.1. In Section 5.1 we talk about transforming this seven-point scale into a binary scale of sensitivity. When we use a discretization that defines all ratings from 4-7 to be sensitive, then the distribution of classes is such that 22% of tweets are sensitive.

From Figure 5.2 we can get some insights that are not unique to our data, but interesting to us nonetheless. E.g. we see that tweets are shorter but more opinionated around noon.

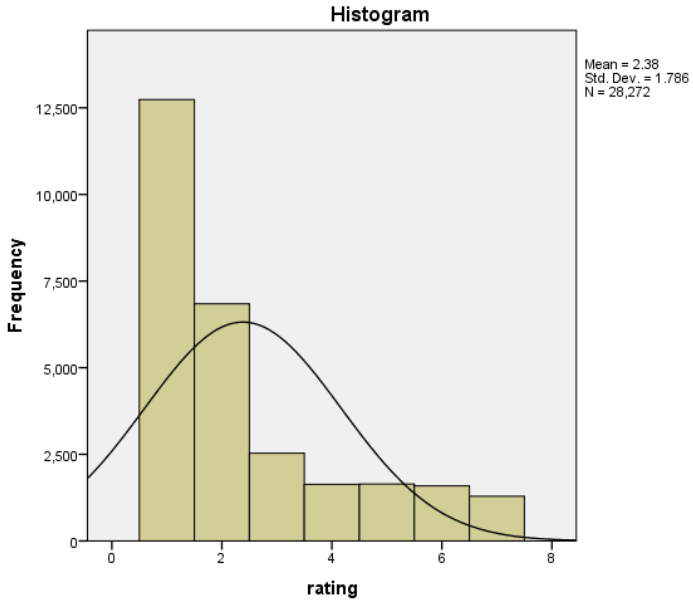
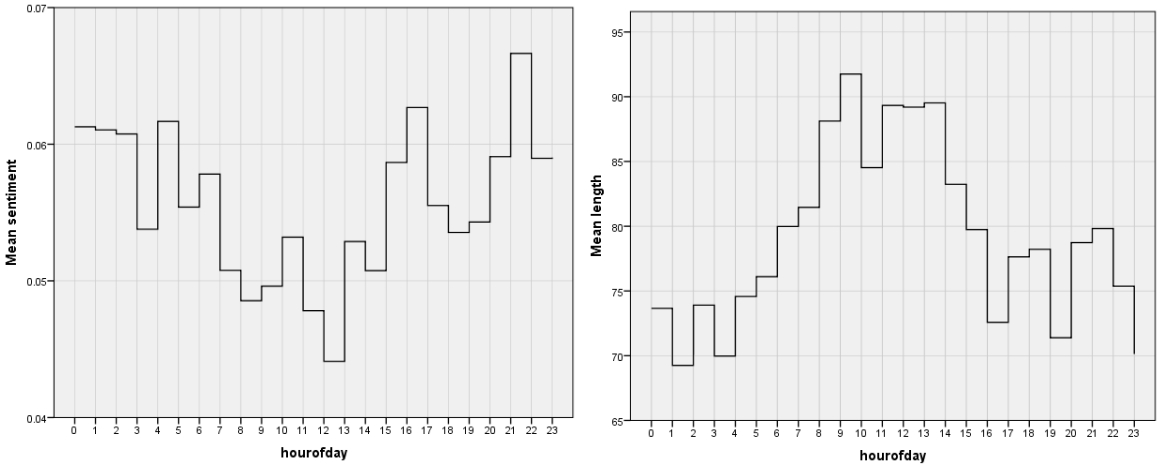


Figure 5.1: Histogram of the distribution of sensitivity judgments.



(a) Mean sentiment score by time of day.

(b) Mean tweet length by time of day.

Figure 5.2: Features that vary over the time of day.

5.3 Predictive Power of the Controversy Score

In Section 3.1.4 we evaluated the mapping of tweets to topics. From these results we took all successful mappings ($grade > 5$) to see how well controversy scores of Wikipedia topics can predict the judged sensitivity rating. In Table 5.1 we show the full list of tweets that were successfully mapped. The topic mentioned in this table is the topic that was the most relevant out of the four generated topics by our multiple topic mapping methods. The most relevant topic was manually selected, therefore this table should not be interpreted as an indication to how well the topic mapping works. Tweets with a controversy score of 0 have a mean rating of 1.81, while tweets with a controversy score higher than 0 have an average rating of 2.27. This does indicate that the controversy score might work as a predictor, but looking at figure 5.3 we see that it is difficult to make predictions based only on controversy score. The figure includes a fitted regression line, but it has a very low R^2 of 0.0051, indicating that it fits badly with the data.

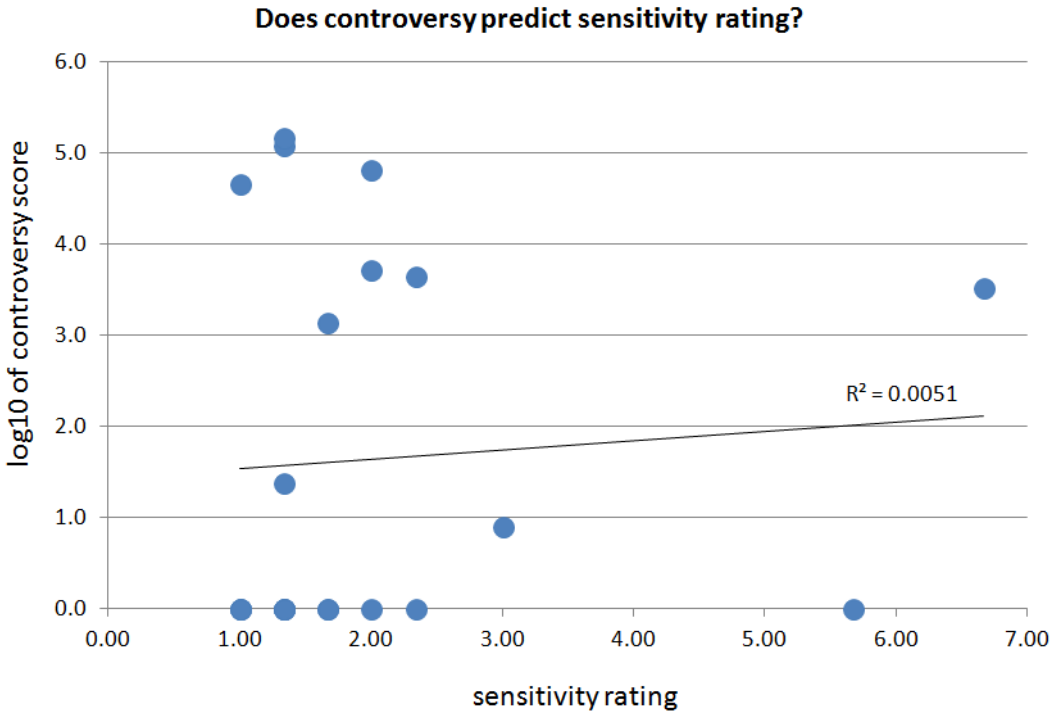


Figure 5.3: A scatter plot of $\log_{10}(controversyscore)$ vs $sensitivityrating$

Tweet text	Topic	Rating	Controversy
I remember Everyone use to be into soccer	Association football	1.00	45603
PanicAtTheDisco pretty pretty please	Panic! at the Disco	1.33	119140
I wanna go to the shooting range	Shooting	1.00	0
BroncosItaly If there s 1 team snake bitten when it comes to the playoffs the Capitals rank high up there	Playoffs	1.33	0
You know that public education is failing when high school seniors think London is its own country	London	1.33	146268
Nine Inch Nails Something I can never have still v a YouTube	The Day the World Went Away	2.33	0
sighalectics I m putting a root vegy cassoulet on the restaurants menu	Cassoulet	1.00	0
Caution for jamiemcmurray Who was inside the Top 5 Josh_Wise ClintBowyer and MichaelAnnett No 41 leads with 25 to go NASCAR	NASCAR	2.00	65720
Omg is this the Gaddafi y all always go on about shhhhhhiiiiiiiitttttttt	Muammar Gaddafi	5.67	0
Dominating possession 70	Ball possession	1.67	0
Deals 318 http t co kKesDP0zYo ARCTIC SILVER 5 AS5 3 5G CPU Thermal Compound Paste Grease 3 5g Tube lot of 2	Thermal grease	2.00	0
Comment on Harry Styles Gets A Mysterious New Tattoo See The Pic by patricia goden I tink it ff bobbypindas	Tattoo	2.00	5252
NowPlaying Cash Cash Surrender EDM ChicagoMusic GenY	Cash Cash	1.00	0
Rainbow Loom Rubber Band Bracelets Full read by eBay	Rainbow Loom	1.67	0
Hey Brady This is Tshirt amp Hoodies Hoodie was designed with Brady BUY Now GenesisHomesQld	Hoodie	1.67	1392
1st day in my new position I m anxious as hell Fear of the unknown	Fear	3.00	8
Yottaa How Well Do You Understand eCommerce Mobile Performance	E-commerce	1.33	0
Starting Steven Gerrard made sense but Liverpool s captain can no longer seize the big occasion Mirror Football	Steven Gerrard	2.33	4444
Sitting down to read Harry Potter and the Goblet of fire This is one of the things I enjoy most about having no show	Harry Potter and the Goblet of Fire (film)	1.33	24
NowPlaying Led Zeppelin Fool In The Rain	Fool in the Rain	1.33	0
Eichel Understands Murray s disappointment says however I think I d be a great teammate hockey	Jack Eichel	1.33	0
Play pussy and get fucked	Pussy	6.67	3255

Table 5.1: Tweets mapped to topics, mapped to controversy score, compared to sensitivity rating. Only tweets with a good topic mapping are included. All URLs and non-alphanumeric characters are removed.

5.4 Parameter & Feature Selection

We have made some groups of features to determine which features are the most useful. When we talk about numerical features we mean features that are represented as a single number, rather than a category. Almost all features are numerical, except for *country, source, lang*, and *weekday*. Tweet-specific features are features that relate to the individual tweet, as opposed to features that relate to the author of the tweet. The following features are considered tweet-related: *reply, coordinates, symbols, hashtags, user_mentions, urls, retweet_count, timeofday, length, sentiment, controversy, language_quality*, and *possibly_sensitive*.

5.4.1 Correlations

Figure 5.7 shows how some of the features correlate with the sensitivity score. Figure 5.6 shows how the sensitivity rating relates to the standard deviation of the sensitivity rating. Figure 5.4 shows the mean sentiment rating for countries for which we have more than 500 judgments. These countries represent 22327 judgments, or 79% of our total data set. We see that people in Vietnam give the highest sensitivity rating on average, whereas people in English speaking countries give the lowest sensitivity ratings.

In Figure 5.5 we see how the mean sensitivity rating varies based on the time of day the tweet was originally posted.

5.4.2 F1-Scores

The set of features with the best performance is the set that contains all numerical features, plus the *country* feature. The classification report of a Naive Bayes classifier with these features is shown in Table 5.3. Table 5.2 shows the results for all features. Table 5.5 show the results with just sentiment as a feature. Table 5.4 shows the results for only all tweet-specific features. That is: reply, coordinates, symbols, hashtags, user_mentions, urls, retweet_count, timeofday, length, sentiment, controversy, language_quality, and possibly_sensitive. See Table A.1 in the appendix for more combinations of features. All precision, recall, and f1 scores in this section and in the appendix are calculated based on a data set that is balanced to have a 50/50 class distribution.

5.4.3 ROC-Curves

We plotted the ROC-curves for a number of combinations of parameters and features. These can be seen in Figures 5.8 and 5.9. ROC-curves show the true positive rate against the false positive rate. These curves were created using six-fold cross validation.

	precision	recall	f1-score	n
notsensitive	0.60	0.38	0.46	6159
sensitive	0.55	0.75	0.63	6159
avg/total	0.57	0.56	0.55	12318

Table 5.2: Gaussian Naive Bayes. All features.

	precision	recall	f1-score	n
notsensitive	0.57	0.56	0.56	6159
sensitive	0.57	0.57	0.57	6159
avg/total	0.57	0.57	0.57	12318

Table 5.3: Gaussian Naive Bayes. All numerical features + country.

	precision	recall	f1-score	n
notsensitive	0.53	0.58	0.55	6159
sensitive	0.54	0.49	0.51	6159
avg/total	0.53	0.53	0.53	12318

Table 5.4: Gaussian Naive Bayes. Tweet-specific features.

	precision	recall	f1-score	n
notsensitive	0.55	0.50	0.53	6159
sensitive	0.54	0.59	0.56	6159
avg/total	0.55	0.55	0.55	12318

Table 5.5: Gaussian Naive Bayes. Only sentiment score as feature.

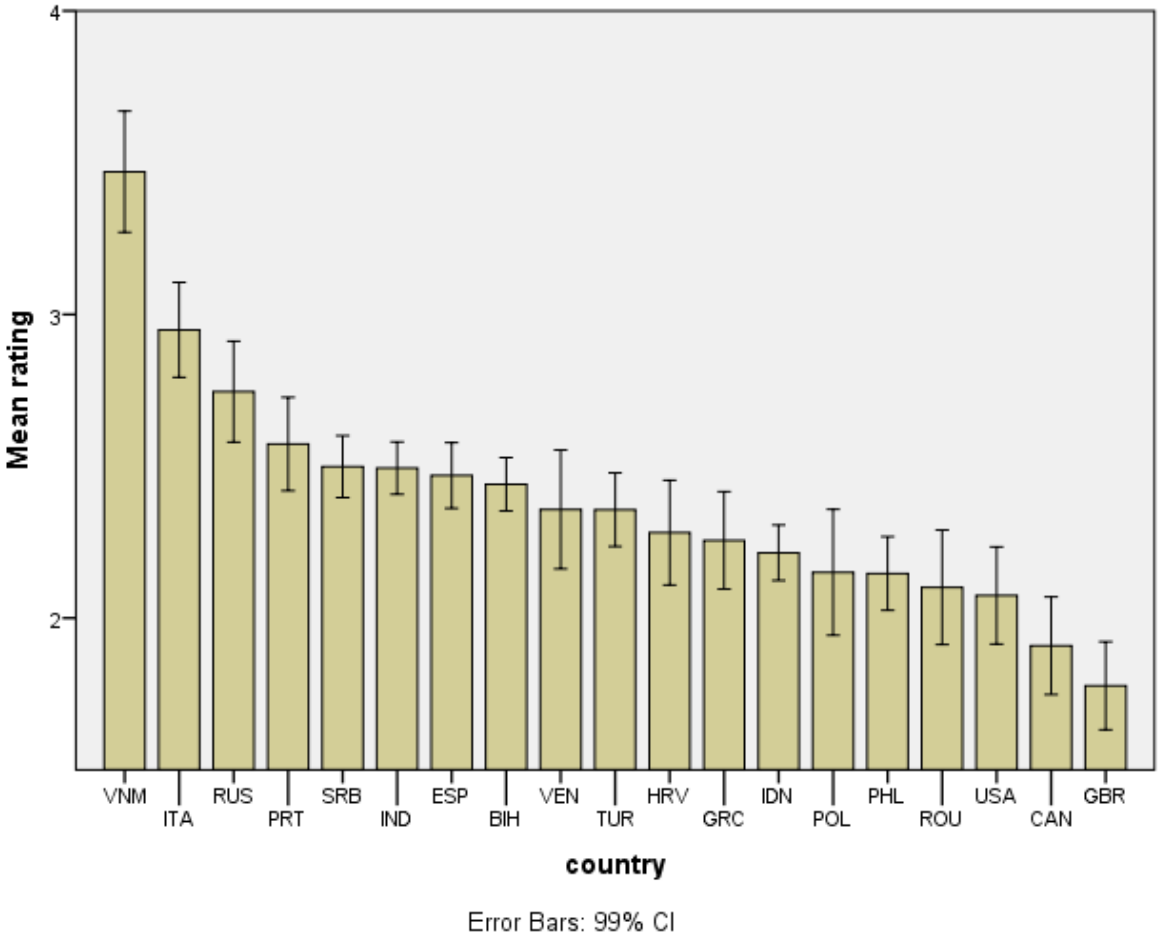


Figure 5.4: The mean rating of judgments from each country. Only countries with more than 500 judgments and more than 6 workers are included.

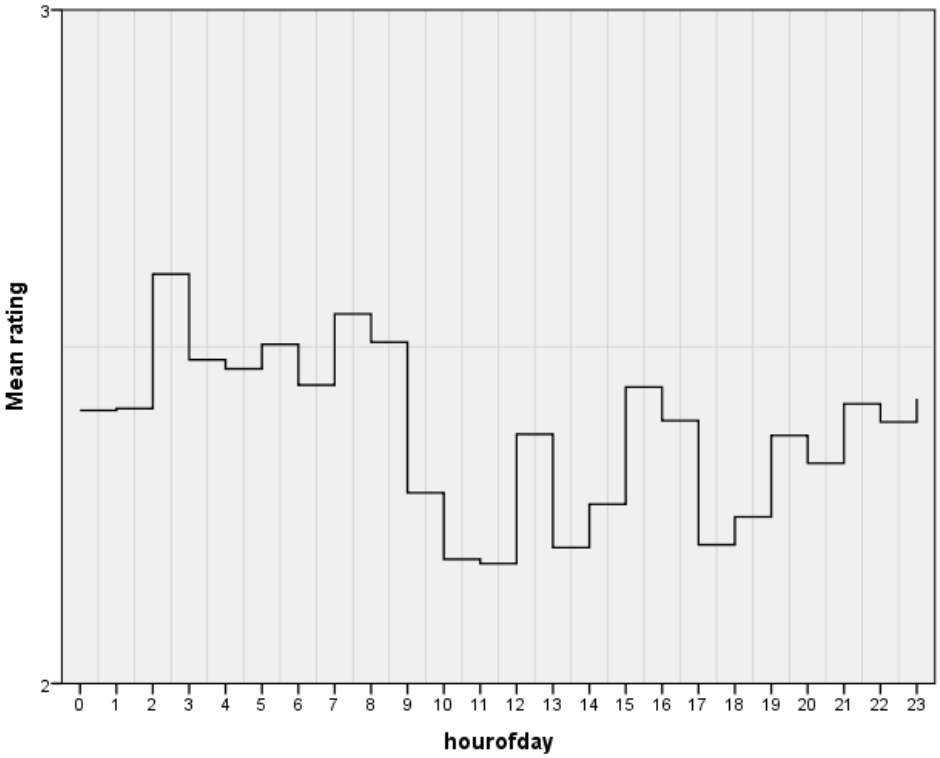


Figure 5.5: Mean sensitivity rating by time of day.

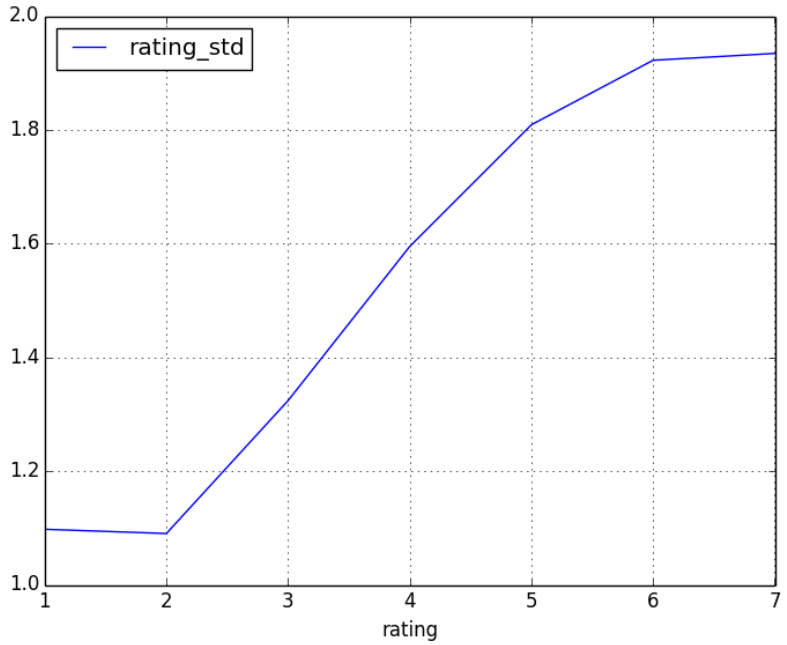
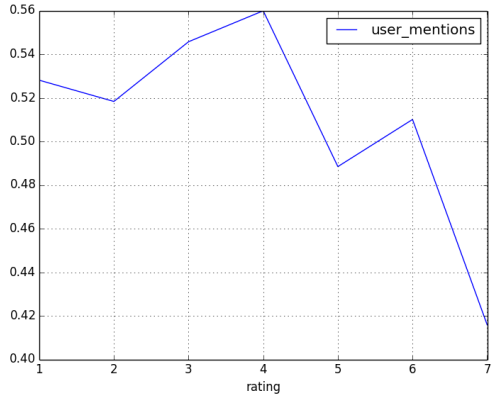
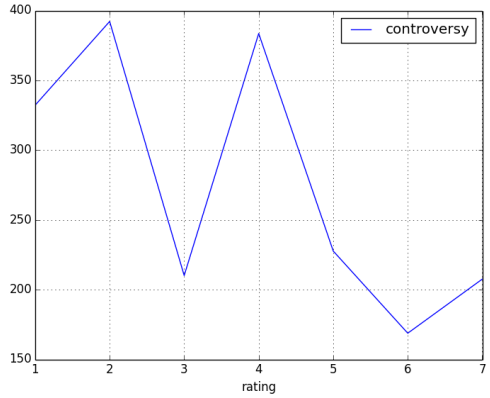


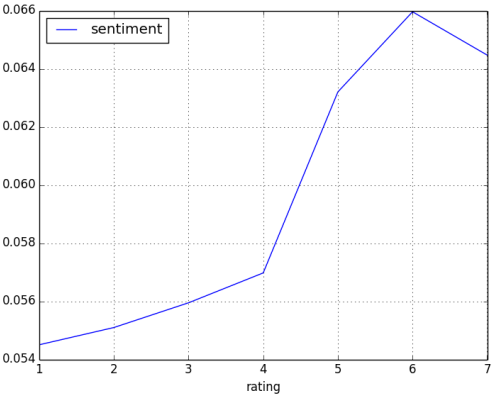
Figure 5.6: Sensitivity rating plotted against the standard deviation of the sensitivity rating.



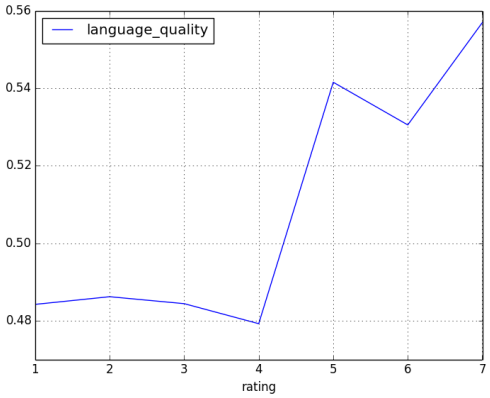
(a) Number of @-mentions



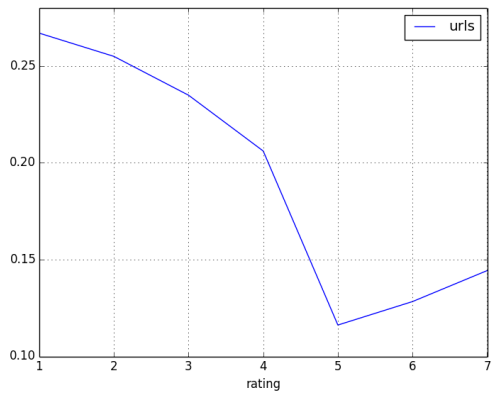
(b) Controversy score



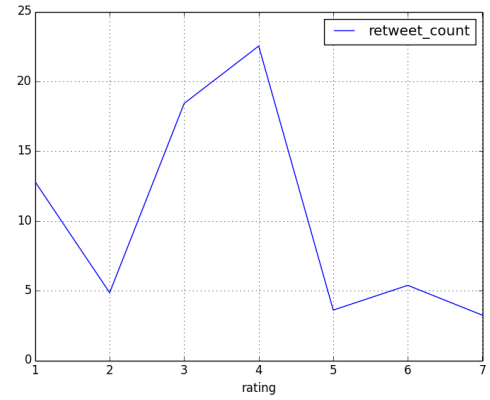
(c) Sentiment (subjectivity) score



(d) Language quality score



(e) Number of urls



(f) Amount of retweets

Figure 5.7: Sensitivity rating plotted against other variables.

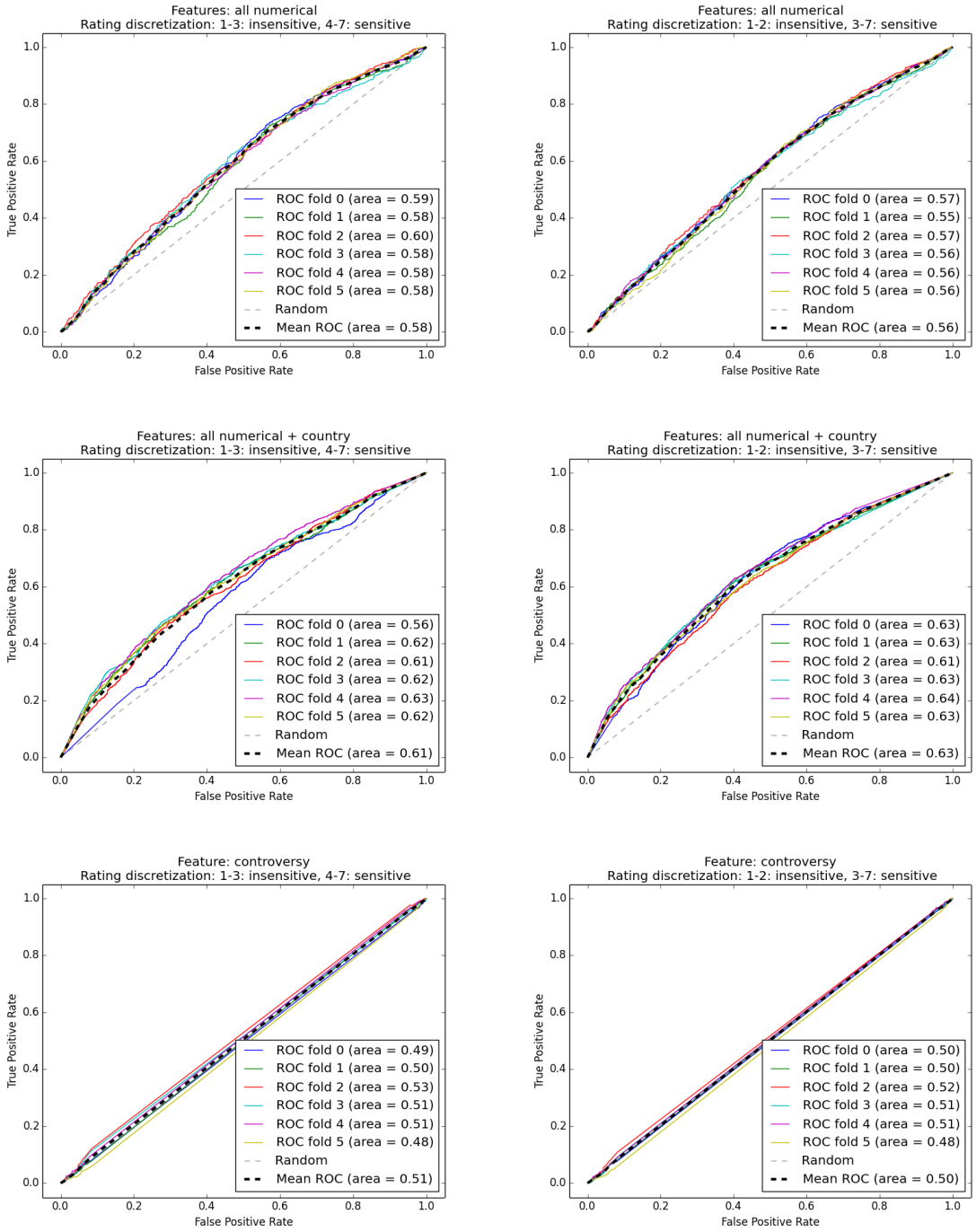


Figure 5.8: ROC-curves for different sets of features, and two different ways of discretization.

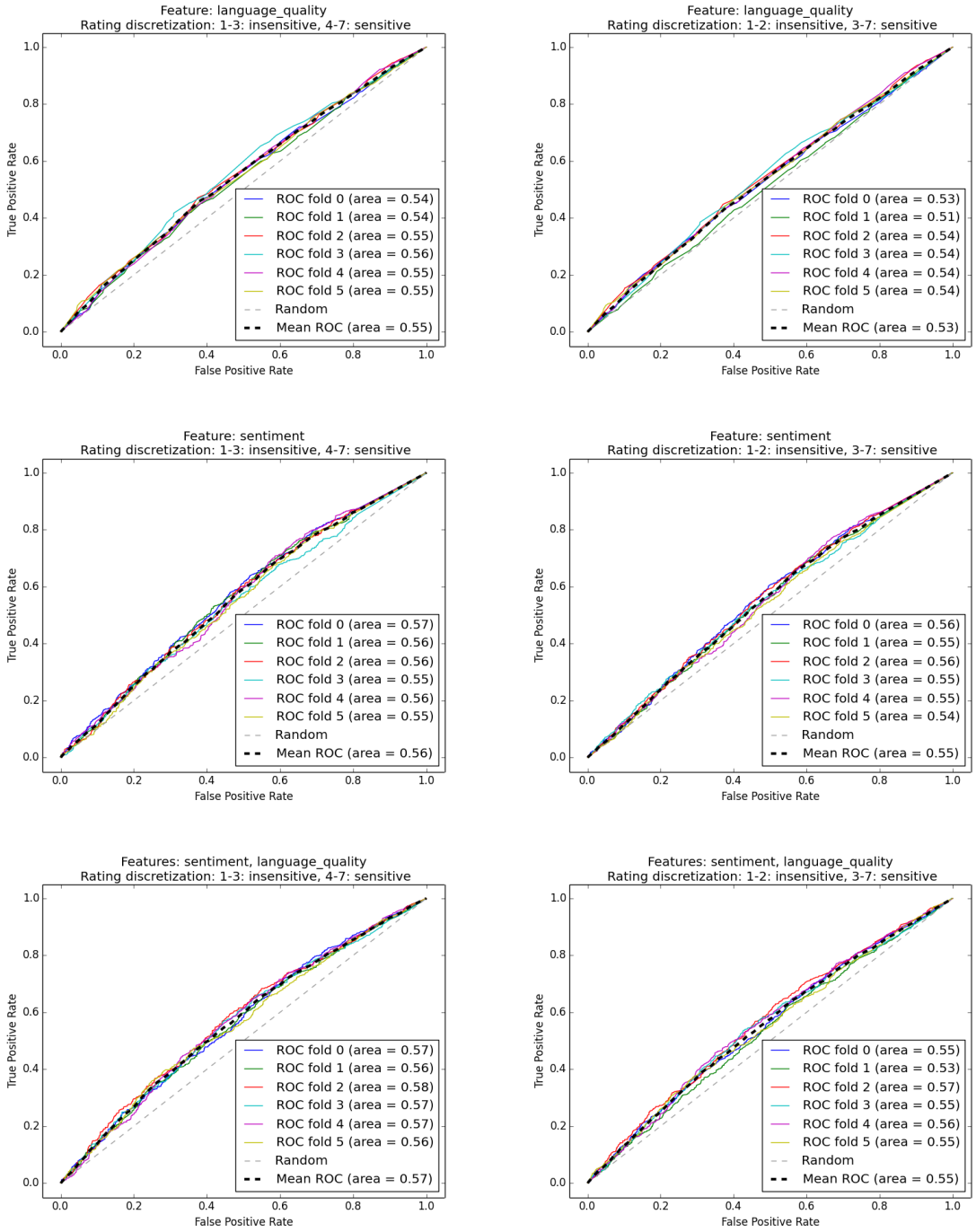


Figure 5.9: ROC-curves for different sets of features, and two different ways of discretization.

5.5 Classification Algorithm Selection

We tested the performance of four classification algorithms with sentiment and language quality as features. These algorithms, and their respective mean F1-scores are the following: Naive Bayes (0.55), Support Vector Machines (0.54), Decision Tree Learning (0.42), and Nearest Neighbors (0.45). The Naive Bayes classifier performed the best, so it is the one used throughout the rest of the analysis. Figure 5.10 show a partial visualization of a decision tree classifier.

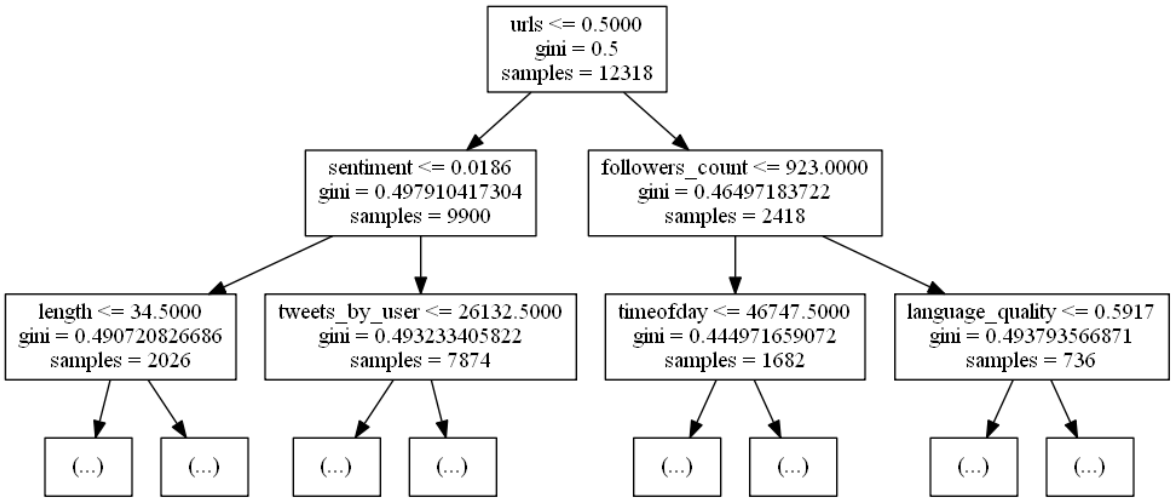


Figure 5.10: A partial visualization of a decision tree classifier

5.6 Objective vs Subjective Sensitivity

A problem with predicting the sensitivity of a message is that it can be subjective. Different people can have different privacy preferences. Ackerman et al. [3] conducted a survey of 381 US web users to find out which privacy concerns those people have. They clustered people into three groups, in order of most to least privacy-concern: privacy fundamentalists (17% of respondents), pragmatists (56%), and marginally concerned(27%). Based on the results of the survey, Ackerman et al. recommend to not use a one-size-fits-all approach.

This subjectivity could be the cause that our classifier is not performing as well as we hope. Therefore it useful to look at how an individual classification model compares to a global model. In addition, there might be groups of people who have similar opinions on which message is sensitive and which isn't. We hypothesize that such a group of like-minded people might be formed by grouping people from the same country.

To compare individual, global, and group classification models, we used the data from 250 of the workers who labeled the tweets in our data set. These 250 workers were specifically selected because they were the ones who labeled a total of 96 tweets each. That was the maximum allowed number of tweets any single worker was allowed to label.

Table 5.6 shows the area under the ROC-curve for individual, global, and group classification. Global classification uses validation data from one person and uses training data, randomly selected from the global data set, of the same size as the validation data. Individual

classification also uses validation data from one person, but uses only the training data from that one person. The group classification uses training data randomly selected from all people of the same country as the person. When there are no people from the same country, the value in this table is 0. The validation data is not used as training data.

We can not draw much conclusions from these results. The mean of each classification type is around 0.50, indicating that the performance is the same as random chance. This is likely due to the tiny amount of training data that we have per individual.

Person	indiv	global	group
1	**0.54**	0.42	0.47
2	0.40	**0.65**	0.59
3	0.53	0.53	**0.56**
4	0.47	0.62	**0.67**
5	**0.55**	0.45	0.54
6	**0.65**	0.46	0.32
7	0.53	**0.55**	0.50
8	**0.71**	0.47	0.00
9	0.41	0.49	**0.64**
10	0.49	0.55	**0.58**
...	...	...	...
250	0.48	**0.58**	0.00
Mean	0.51	0.52	0.51
Sum	85	97	79

Table 5.6: Area Under ROC-curve for individual, global, and group classification. The best classification type is presented in bold. The *sum*-statistic represents the number of times that the classification type was the best one out of the three.

Chapter 6

Related Work

Sriram et al. [16] showed that traditional bag-of-words classifiers are outperformed by a classifier with better selected features, when it comes to classifying short texts (tweets in this case) in to classes such as news, events, and opinions. The features suggested by Sriram et al. are: author, abbreviation use, time-event phrases, opinion words, emphasis on words, currency and percentage signs, and @-mentions.

Wang et al. [19] experimented with giving privacy nudges to Facebook users. They tested three types of nudges: A picture nudge which showed the profile pictures of a random selection of the people who the post was about to be shared with, a timer nudge which gave the user the possibility to undo the post up to 10 seconds after posting, and a sentiment nudge. The sentiment nudge calculates the overall sentiment of the post, and shows the result (e.g. "your post is very negative") to the user. The sentiment nudge was tested with eight user study participants, half of which were interviewed. The results are therefore qualitative in nature, but seem to indicate that the nudges help people to better consider their posting behavior. The picture nudge and the timer nudge were thought to work better than the sentiment nudge. The authors acknowledge the fact that a quantitative study would be interesting in order to validate the idea.

Machida et al. [10] are one of the few people that we know of that are working on a system for detecting sensitive information in social network posts. They use keyword analysis and semantic orientation analysis with a support vector machine classifier. The classification occurs along two axis:

- The category of the content (religion, beliefs, medical history, mental records, photos, criminal behavior, domestic situations and personal behavior).
- The degree of importance of the information (split in three levels).

Machida used the degree of importance to recommend a specific social group that the information is suitable for. This works well if these social groups are defined, but with a lot of online social networks these groups are typically unknown. Mondal et al. [12] found that it is also very difficult to predict these groups automatically, because there is very little correlation between profile information and being in certain social group.

The most promising research in predicting regrettable messages on social networks comes from Zhou et al. [21]. They use deleted tweets as examples of tweets that are regrettable. their machine learning algorithm uses features such as sentiment and curse words. The

best performing algorithm (J48) produces an F1-score of 0.849, which indicates that a good classification is achieved.

Chapter 7

Conclusion

This study has shown that using a Naive Bayes classifier with a specific set of features is a promising approach to predict whether a message on a social network is regrettable or not. The feature that are useful are sentiment, language quality, time of day, number of urls. Amount of retweets and controversy score are not useful. Although about the controversy score we should remark that we don't have a good way yet to accurately calculate the controversy score of a piece of text.

Our hypothesis that a high sentiment score results in high sensitivity is correct, but the same cannot be said for the controversy score. The amount of retweets is actually the lowest for the tweets that are the most sensitive, so our hypothesis about that was false. The same goes for the amount of @-mentions. With more mentions we generally see lower sensitivity. We saw that the standard deviation of the judged sensitivity gets higher when the judged sensitivity itself gets higher, so we can confirm hypothesis 5. We do not have enough data to answer the hypothesis about whether an individual model works better than a global model.

The research findings have brought us a step closer to finding a way to classify social network posts as either sensitive or not. If this research direction is further investigated it could result a number of different applications to improve the online privacy of social network users.

Our study has some major limitations. First, the data we collected does not represent real regret that users have about posting messages online. Because the data was gathered by crowdsourcing, it actually represents what people think that other people might regret posting. This is a difference. The question that we asked to the crowdsourced workers is also open for interpretation. Secondly, our data set was small. We didn't have enough data for instance to come to a good conclusion about whether individual models work better than global models.

7.1 Future Work

Future works should look into ways of finding more and different training data. There might be large amounts of data out there that could serve as a proxy for finding the regrettability of a message. We also expect that social networking companies like Twitter, Facebook or Google have the right kind of data, which includes both public messages as well as private messages. We could imagine that comparing the private messages against the public messages on these websites would result in a good way to predict whether unlabeled messages should

be private.

In the introduction we mentioned a potential application as a tool to clean up your social network account before going on a job interview, using a system that automatically detects potentially sensitive messages. Such a system could propose a few public messages that are recommended to be deleted. The user may choose whether to agree to delete each message or not. This decision is an interesting data point. We could ask the user to allow us to anonymously use the deleted and not deleted messages as training data. With enough users of this tool this can be a very good way to find large amounts of very relevant training data.

If the future work chooses to gather training data through crowdsourcing, then it is recommended to look at smart ways to choose the messages to be labeled. Most messages found online are not sensitive, so this class was underrepresented in our study.

The controversy score did not look so promising in our study, but this might be the case because we were very bad at mapping short texts to topics. However, the field of Named Entity Recognition has a similar goal of mapping text to semantic entities. Techniques used in this field might prove useful to find a good controversy score.

There is still a lot more information contained in a social network message than we have discussed. Messages often contain links to webpages, which could themselves contain sensitive content. Messages can also contain pictures and videos, which makes the problem even more complex by introducing the need for computer vision.

Bibliography

[1] Apache solr. `https://lucene.apache.org/solr/`. Accessed: 2015-06-29.

[2] Yahoo query language. `https://developer.yahoo.com/yql`. Accessed: 2015-05-17.

[3] Mark S Ackerman, Lorrie Faith Cranor, and Joseph Reagle. Privacy in e-commerce: examining user scenarios and privacy preferences. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 1–8. ACM, 1999.

[4] Alessandro Acquisti and Jens Grossklags. Privacy and rationality in individual decision making. *IEEE Security & Privacy*, 2:24–30, 2005.

[5] Alessandro Acquisti and Jens Grossklags. What can behavioral economics teach us about privacy. *Digital privacy*, page 329, 2007.

[6] Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 295–303. Association for Computational Linguistics, 2010.

[7] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422. Citeseer, 2006.

[8] Yegin Genc, Yasuaki Sakamoto, and Jeffrey V Nickerson. Discovering context: classifying tweets through a semantic transform based on wikipedia. In *Foundations of Augmented Cognition. Directing the Future of Adaptive Systems*, pages 484–492. Springer, 2011.

[9] Hugo Liu, Henry Lieberman, and Ted Selker. Automatic affective feedback in an email browser. In *In MIT Media Lab Software Agents Group*, 2002.

[10] Shimon Machida, Tomoko Kajiyama, Shigeru Shimada, and Isao Echizen. Poster: Adaptive disclosure control system using detection of sensitive information in snss.

[11] Naresh K Malhotra, Sung S Kim, and James Agarwal. Internet users' information privacy concerns (iuipc): the construct, the scale, and a causal model. *Information Systems Research*, 15(4):336–355, 2004.

[12] Mainack Mondal, Yabing Liu, Bimal Viswanath, Krishna P Gummadi, and Alan Mislove. Understanding and specifying social access control lists. In *Symposium on Usable Privacy and Security (SOUPS)*, 2014.

[13] Soumya Ranjan Patra, Mykola Pechenizkiy, and Erik Tromp. Named entity recognition and disambiguation in tweets. 2014.

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

[15] Manya Sleeper, Justin Cranshaw, Patrick Gage Kelley, Blase Ur, Alessandro Acquisti, Lorrie Faith Cranor, and Norman Sadeh. i read my twitter the next morning and was astonished: a conversational perspective on twitter regrets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3277–3286. ACM, 2013.

[16] Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 841–842. ACM, 2010.

[17] Róbert Sumi, Taha Yasseri, András Rung, András Kornai, and János Kertész. Edit wars in wikipedia. *arXiv preprint arXiv:1107.3689*, 2011.

[18] Yang Wang, Pedro Giovanni Leon, Xiaoxuan Chen, Saranga Komanduri, Gregory Norcie, Kevin Scott, Alessandro Acquisti, Lorrie Faith Cranor, and Norman Sadeh. The second wave of global privacy protection: From facebook regrets to facebook privacy nudges. *Ohio St. LJ*, 74:1307–1335, 2013.

[19] Yang Wang, Pedro Giovanni Leon, Kevin Scott, Xiaoxuan Chen, Alessandro Acquisti, and Lorrie Faith Cranor. Privacy nudges for social media: an exploratory facebook study. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 763–770. International World Wide Web Conferences Steering Committee, 2013.

[20] Yang Wang, Gregory Norcie, Saranga Komanduri, Alessandro Acquisti, Pedro Giovanni Leon, and Lorrie Faith Cranor. I regretted the minute i pressed share: A qualitative study of regrets on facebook. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*, page 10. ACM, 2011.

[21] Lu Zhou, Wenbo Wang, and Keke Chen. Identifying regrettable messages from tweets. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 145–146. International World Wide Web Conferences Steering Committee, 2015.

Appendix A

Appendix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Predicted sensitive(%)	.95	.45	.84	.66	.51	.78	.69	.69	.22	.47	.71	.71	.40	.52	.54	.48	.89	.58	.80	.11	.99
Accuracy	.51	.53	.54	.53	.57	.56	.56	.56	.55	.56	.56	.56	.57	.55	.55	.54	.50	.52	.55	.50	.50
Average f1	.39	.53	.48	.52	.57	.52	.55	.55	.51	.56	.54	.55	.56	.55	.55	.54	.41	.51	.51	.41	.35
Average precision	.58	.53	.57	.54	.57	.59	.57	.57	.58	.56	.57	.58	.57	.55	.55	.54	.49	.52	.58	.50	.57
Average recall	.51	.53	.54	.53	.57	.56	.56	.56	.55	.56	.56	.56	.57	.55	.55	.54	.50	.52	.55	.50	.50
Not sensitive f1	.12	.55	.30	.44	.56	.39	.46	.46	.65	.57	.45	.45	.61	.54	.53	.55	.17	.47	.36	.64	.04
Not sensitive precision	.64	.53	.63	.55	.57	.63	.60	.60	.53	.55	.61	.61	.55	.55	.55	.54	.49	.52	.63	.50	.64
Not sensitive recall	.07	.58	.20	.37	.56	.28	.38	.38	.83	.59	.36	.36	.67	.53	.50	.56	.10	.44	.25	.90	.02
Sensitive f1	.67	.51	.66	.60	.57	.66	.63	.63	.38	.54	.64	.64	.52	.56	.56	.53	.64	.55	.66	.18	.67
Sensitive precision	.51	.54	.52	.52	.57	.54	.55	.55	.62	.56	.54	.55	.58	.55	.54	.54	.50	.51	.53	.51	.50
Sensitive recall	.96	.49	.88	.70	.57	.84	.75	.75	.27	.52	.77	.77	.46	.58	.59	.52	.89	.60	.86	.11	.99
_country					y	y	y	y	y				y								
controversy		y	y	y	y	y	y	y	y								y				
coordinates		y		y	y	y	y	y	y												
favorite_count	y		y	y	y	y	y	y	y												
follow_ratio	y		y	y	y	y	y	y	y												y
followers_count	y		y	y	y	y	y	y	y												
following_count	y		y	y	y	y	y	y	y												
hashtags		y	y	y	y	y	y	y	y												
lang							y	y	y												
language_quality		y	y	y	y	y	y	y	y		y	y	y	y		y					
length		y	y	y	y	y	y	y	y												
possibly_sensitive		y		y	y	y	y	y	y												
reply		y	y	y	y	y	y	y	y												
retweet_count		y	y	y	y	y	y	y	y	y										y	
sentiment		y	y	y	y	y	y	y	y	y	y	y	y	y	y						
source						y	y	y													
symbols		y		y	y	y	y	y	y												
timeofday		y	y	y	y	y	y	y	y		y							y			
tweets_by_user	y		y	y	y	y	y	y	y												
urls		y	y	y	y	y	y	y	y	y	y	y							y		
user_mentions		y	y	y	y	y	y	y	y												
weekday								y	y												

Table A.1: Gaussian Naive Bayes. 50/50 balanced class distribution.