Troll detection with sentiment analysis and nearest neighbour search

FILIP JANSSON
OSKAR CASSELRYD
Troll detection with sentiment analysis and nearest neighbour search

Oskar Casselryd, Filip Jansson

VERSION 1.0
FINAL VERSION

June 5, 2017
Abstract

Internet trolls are gaining more influence in society due to the rapid growth of social media. A troll farm is a group of Internet trolls that get paid to spread certain opinions or information online. Identifying a troll farm can be difficult, since the trolls try to stay hidden. This study examines if it is possible to identify troll farms on Twitter by conducting a sentiment analysis on user tweets and modeling it as a nearest neighbor problem. The experiment was done with 4 simulated trolls and 150 normal twitter users. The users were modeled into data points based on the sentiment, frequency and time of their tweets. The result of the nearest neighbor search could not show a clear link between the trolls as their behaviour was not similar enough.
Identifiering av troll med sentiment analys och nearest neighbour search.

Sammanfattning

## Contents

1 Introduction ........................................ 1
   1.1 Problem statement ............................... 2
   1.2 Scope ........................................... 2

2 Background ......................................... 3
   2.1 Twitter ........................................... 3
   2.2 Twitter API, limitations and policies .......... 3
   2.3 Troll farm ....................................... 4
       2.3.1 Troll distinction ............................. 4
       2.3.2 Farm ........................................ 4
   2.4 Sentiment Analysis ............................... 5
       2.4.1 Levels of investigation ..................... 5
       2.4.2 Sentiment analysis on tweets ................ 6
       2.4.3 Stanford coreNLP ............................. 6
   2.5 Distance metrics ................................ 6
       2.5.1 Euclidean metric ............................ 6
   2.6 Hamming metric ................................ 7
   2.7 Nearest neighbour search ......................... 7
       2.7.1 Exact NNS, variations and problems ........ 7
       2.7.2 Approximations ............................... 8
       2.7.3 Locality-Sensitive Hashing ................. 8

3 Method .............................................. 10
   3.1 Data set .......................................... 10
   3.2 Choosing parameters ............................. 11
   3.3 Sentiment tools .................................. 12
   3.4 Nearest neighbour implementation ............... 13

4 Results ............................................ 14
   4.1 Troll data ....................................... 14
   4.2 Troll relative closeness ......................... 15
   4.3 Comparison trolls and normal users ............. 15
5 Discussion
  5.1 Results .......................................... 20
  5.2 Method ............................................ 21

6 Conclusion ........................................... 22
1 Introduction

Large quantities of opinions and diverse thoughts flow through social media every day. It serves as a platform for people to share thoughts on hot topics and conduct public debate. The general sentiment on social media can affect many peoples opinion and comprehension of different topics. Twitter has over 300 million monthly users [12], and with the growing user base a number of professional "trolls" have been revealed on the platform [1]. It has become increasingly more important to identify these users as they often represent imaginary people with made up opinions, thus affecting the debate by giving an impression of a wider support or opposition for selected topics. There are many different definitions and uses of the word "troll". In a modern context it often refers to someone sitting at home spewing hatred or disinformation through different forums, often anonymously[3].

A new phenomenon are so called "troll farms". It refers to a professional setting where people get paid to troll on account of their employer. Just like a normal business they work shifts behind a screen in their cubicles. Many popular platforms take actions against serious trolls, much like they do against bots spamming the network. To avoid detection the people at troll farms often operate from several different accounts and use software that keeps their location hidden. This activity has similarities with opinion spamming, which often is done by bots with the intention of drowning feeds with useless information. Thus making it hard to find relevant information. The difference between trolling and opinion spamming is that trolls try to give a higher sense of relevance in their text.

There are many ways to detect users with similar behavioural patterns, but they are often expensive and therefore not realistic to implement. Most methods depend on modelling users as vectors or points in a coordinate system. This makes it easier to
compare similarities. Ways of obtaining data for these models vary. Information such as time and location can often be extracted from the users social media profile. The data can then be used directly in the model or put through an analytic process to convert it to a more appropriate format.

The field of analyzing human-written text with computational means is called Natural Language Processing. It has many sub-fields, one of them being sentiment analysis. There are variations in how sentiment analysis is conducted but the goal is usually to quantify the sentiment of a text. The output of such a process could be single number.

1.1 Problem statement

This study aims to answer the following question: Is it possible to identify troll farms on Twitter by conducting a sentiment analysis on user tweets and modeling it as a nearest neighbour problem?

1.2 Scope

The study was conducted on a set of tweets from 150 randomly selected Twitter profiles together with 4 simulated trolls over a time period of 4 days. This report focuses on the data gathered from the tweets rather than the users. This decision was made to keep the focus of the report on sentiment analysis. A more detailed account of the experiments can be found in chapter 3.
2 Background

This chapter provides a brief background to the methods, tools, platform and concepts central to the study.

2.1 Twitter

Being one of the most popular micro blogs in the world, twitter allow users to post messages up 140 characters long, referred to as "tweets". Additional links and "hash tags" can be attached to the tweets. Hash tags are labels that can be used to find messages concerning certain topics.

2.2 Twitter API, limitations and policies

The twitter-API give developers a way to interact and use the data available on the site. The API provides a powerful search tool with many parameters for writing queries. It also presents a time-line tool that can be used to generate the most recent tweets from a certain user. It has a rate limit of 100 calls per fifteen minutes. This makes tweet extraction of larger sets of tweets time consuming.

The different API tools have some built-in constraints put in place to protect the privacy of twitter users. The search device has a time-based restriction, only returning tweets that were published within seven days of the API call. In contrast to that the time line tool exists. It is only limited by rate, returning a maximum of two hundred tweets.

Twitter does not allow third party hosting of publicly available databases containing tweets. This means that every set of tweets gathered by a third party only contains unique tweet-IDs. These IDs can be used to extract additional information about said tweets through the official API. The intent of this policy is
to preserve the users right to delete tweets, and for twitter to remain in control of the data. [13]

The process of extracting tweets from public data sets of IDs is referred to as ”hydrating”. Several unofficial, but reliable, tools for this purpose are available. The hydration is done through GET requests to the twitter API’s, returning a JSON object with the all twitter data regarding the object.

2.3 Troll farm

In order to understand what a troll farm is, the concept of a ”troll” must first be specified.

2.3.1 Troll distinction

The word troll has many different meanings and nuances. In modern times it usually refers to an internet troll. The official definition of an internet troll is ”someone who leaves an intentionally annoying message on the internet, in order to get attention or cause trouble”[3]. However, the term is sometimes used in a broader sense to describe people with the intention to deceive or spread misinformation online. [1]

2.3.2 Farm

The term ”troll farm” refers to a group of professional trolls. Being professional means they get paid to spread disinformation in favour of their employer. The goal of troll farms is to influence the public opinion and give an illusion of a broader support in favour of their employers interests. For example there have been reports about troll farms being linked to the state-sponsored Russian web brigade, which is tasked to spread pro-Russian and pro-putin propaganda on different web-based platforms[6].
The employees at a troll farm are often divided into smaller groups of five to six people. The groups receive tasks each day relating to different topics that they are to influence. They then spread opinions corresponding to instructions about said topic. The employees almost always work under several different online aliases. This is partly to avoid detection but it also serves the goal of giving a sense of more people supporting their cause [1].

2.4 Sentiment Analysis

Also known as ”opinion mining”, sentiment analysis aims to determine whether a piece of text expresses positive, negative or neutral sentiment. It is a subfield of Natural Language Processing (NLP) which is based around problems concerning computational interpretation of human languages [2].

2.4.1 Levels of investigation

Sentiment analysis can be divided into three levels of investigation. Namely document level analysis, with the goal of getting an overall sentiment on an entire document. Sentence level analysis, which has the same purpose, but only for a single sentence. Lastly entity and aspect level analysis, where the goal is to detect not only the sentiment, but the object it is directed at [2]. This study uses tools that can handle all three levels [10], but will focus on sentence level analysis as Tweets are usually not more than one sentence or two due to the limitations on message length. It is not relevant what entity it is directed towards either, as that would introduce problems when choosing appropriate data samples.
2.4.2 Sentiment analysis on tweets

Since tweets are short and usually does not contain many different opinions, sentiment analysis is simpler than it would have been on news articles or other longer pieces of text. On the other hand tweets often contain abbreviations, slang and misspellings which could yield misleading results when conducting the analysis. The difficulty often varies a lot with different topics. Tweets concerning social and political discussions often contain complex expressions with sarcasm and irony, while opinions about products and services usually are pretty straightforward and easier to analyse [2].

2.4.3 Stanford coreNLP

One of the currently most prominent open source toolkits for Natural Language Processing is the Stanford coreNLP. It contains a widespread range of tools that can be used in different types of analyses [10]. One of them is the tool for sentiment analysis.

2.5 Distance metrics

Studying similarity between different data points requires a metric and a metric space to describe the relation between points.

2.5.1 Euclidean metric

The Euclidean metric, also known as Pythagorean distance is the most common metric used. Given a space with \( n \) dimensions the Euclidean metric is defined as

\[
\|x - y\|_2 = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2}
\]  

(1)

where \( x \) and \( y \) are vectors in the \( n \) dimensional space [8].
2.6 Hamming metric

The hamming metric is used when comparing strings or vectors of equal length. The hamming distance between $p$ and $q$ can be defined as the minimum number of operations needed to transform $p$ into $q$. This metric is often used when comparing words or binary vectors. The term 'hamming cube' has come to refer to the hamming space of binary vectors. It is possible to represent every binary vector of length $d$ within the hamming cube of dimension $d$. [7]

2.7 Nearest neighbour search

The nearest neighbor search (NNS) is a commonly used technique for pattern recognition and other type of problems. There are many different variants of the algorithm that are effective in different areas. The core problem is defined as follows:

"Given a set $P$ of points in a $d$ -dimensional space, construct a data structure which given any query point $q$ finds the point in $P$ with the smallest distance to $q". [5]

The following sections will describe variations of the NNS with their up- and downsides.

2.7.1 Exact NNS, variations and problems

The NNS can give more information than just the closest neighbour of each data point. The $\epsilon$ - Nearest neighbor search finds all points within a fixed distance from the query point. The $k$-Nearest neighbour search finds the $k$ nearest neighbour to the query point, this can be used to find which $k$ points have the strongest relation to the query point.

The ”straight forward” implementation of the NNS is a linear
exhaustive search through the set of points finding the closest point. This method yields a time complexity of $O(dn)$, where $n$ is the amount of data points. It works well on smaller sets with a low number of dimensions, but scales poorly with the size and dimension of the set.

The query time of the algorithm can be improved if the data points are preprocessed. A popular solution to this is called *kd-trees*. This method splits the data set into $d$ binary trees based on the points position in each dimension. The data structure allows the average query time to be sub-linear. It does however suffer the "Curse of dimensionality", meaning it encounters problems in higher dimensions, e.g. poor scaling. Studies have shown that a linear search can be more effective than the *kd-tree* method when approaching 10-20 dimensions\[4\]. There is currently no exact algorithm for the Nearest neighbour problem that does not struggle in high dimensions.

### 2.7.2 Approximations

Due to the Curse of dimensionality, alternative methods are generally used over the standard implementation. An approximated version of the algorithm is often used when working in higher dimensions, since it allows querying within polynomial time. The drawback is naturally that the result may not always be precise.

### 2.7.3 Locality-Sensitive Hashing

Locality-Sensitive Hashing (LSH) is an approximate method for a nearest neighbour search. This algorithm maps each point of the data set to buckets, where similar data points (using the euclidean metric) have a high probability to end up in the same buckets.

The algorithm uses two kinds of hash functions with opposite purposes. First and foremost we have the locality-sensitive hash
functions. The purpose of these functions is to have similar points collide, which places them in the same bucket. Each hash function does this by projecting the selected data point on to a random set of coordinates. In practice this is done by converting the coordinates to a hamming cube with dimension $d \times C$, where $C$ is the value of the largest coordinate in the system, and then projecting the point in hamming space on to the coordinate set.

The effectiveness of this algorithm depends on two parameters:

- $l$ - The number of local-sensitive hashfunction used.
- $k$ - Number of coordinate sets used within each hashfunction

The choice of the values $k$ and $l$ are essential for having minimal margin of error as well as sub-linear query time. An optimal value of $k$ creates a large number of hash collisions between ”similar” points but also have a low chance of collision from points that are ”different” to the metric used.

The second hash function used is to assign the points in to a certain bucket based on the result from the first hash function. The purpose of this hashing is to avoid collision, therefore a more standard hash function is often used.

Querying on this data-structure is done by running the locality-sensitive hash functions on the query points and collecting all points that we have stored in those buckets. The set of data point we get from this are of a smaller size, therefore we can perform a linear search on that subset.
3 Method

The following chapter will present the process behind the study. This topic is divided into four parts: data set, choice of parameters and NNS implementation. The purpose of this chapter is to make the project repeatable. It will therefore give the reader a precise description of the methodology and try to explain the thought process behind certain decisions.

3.1 Data set

The study requires a data set with tweets from both trolls and normal users. Proper selection of normal users is important in order to recreate a realistic environment in which the trolls operate. Finding a real up-and-running troll farm would be ideal for this study but is a very difficult task.

Instead a troll farm was simulated using four volunteers and newly created twitter accounts. Every person was handed an account. They used the account to publish tweets under certain rules in order to mimic a troll farms behavior. It was conducted during a time period of four days. The rules for every user during the simulation were as following:

- Tweet during working hours (8-17)
- 6-10 tweets per day
- Every day the user gets a different topic to tweet about
- The users were told to be opinionated according to instructions.
- All tweets must to be in English

These users were meant to represent a small scale troll farm.
When choosing normal users a fully random selection is not optimal. Some users could be inactive or write in different languages that our sentiment analysis tool cannot handle. The users where selected at random from a group that fulfilled predetermined requirements. The requirements were that every user must

- tweet in English
- be active on the platform (limited at 20 tweets per day to avoid extreme users with low relevance to our study)
- be located in a timezone close to Sweden.

In order to fulfill the requirements, the users were generated from a pool of users that wrote English tweets in the proximity of London.

Once the users were selected their tweets where acquired through the twitter API.

3.2 Choosing parameters

Appropriate parameters where extracted from the data set using tools developed from scratch for this specific purpose. These were used as coordinates for the different dimensions in the nearest neighbour search later on. Parameters were chosen with hopes of being able to differentiate the troll users from normal users. They were also chosen with respect to difficulty of falsifying the information. Things like location were therefore omitted. The parameters were:

- Average amount of tweets per day.
- Average time of day for tweets.
- Average sentiment of tweets.
3 METHOD

- Standard deviation of sentiment in tweets.

Sentiment analysis was applied with a hypothesis that trolls express stronger opinions than normal users. Largely because they are told explicitly to have opinions. Normal users can also tweet facts or questions which often are deemed neutral in the analysis. Trolls should therefore have a larger variation of sentiment in their tweets. A sentiment score for each tweet was obtained using the Stanford coreNLP toolkit[10]. For more information see subsection 3.3 Sentient tools. Time of day for the tweets was handled in the form of minutes after midnight.

The majority of parameters ended up as decimal points. That is problematic when enforcing the locality-sensitive hashing which require integers. Another problem that arises is the different magnitudes of the parameters. It has a great effect on the euclidean distance. To solve the first problem and compensate for the second one all parameters where scaled with an appropriate factor and then rounded to the nearest integer. This way a higher rate of accuracy was maintained than if the numbers were to be rounded in their original state.

3.3 Sentiment tools

Two different approaches were taken when determining the sentiment of tweets. The tool used allowed for easy quantifiable sentiment analysis on a single sentence. When applied to a sentence it would give a sentiment score as an integer between 0 and 4. A high value meant the sentence was perceived as positive. A low value meant it was perceived as negative. Sentences with sentiment score 2 were considered neutral. At first the tool was only applied to the longest sentence in the tweet. That sentence would then determine the entire tweets sentiment score. This was because the longest sentence is most likely to hold the main points of the tweet. In a second try, the tool was used on all
sentences in a tweet and the mean of their sentiment score determined the tweets score. Result from both methods are presented in chapter 4.

3.4 Nearest neighbour implementation

Similarity between points was compared with a k-nearest neighbour search. It is a suitable algorithm for detecting a set amount of users with strong relations to the query point. Accordingly it can show whether the trolls are grouped closer to each other than to other users with the given parameters. A positive result would show that this method is successful over the data set.

There are many varieties of NNS that would have been sufficient for this study. In the end the LSH was chosen and implemented. The drawbacks of the algorithm are few with the right settings and it allows for both increased sample size and number of dimensions in case it would become relevant.

Another approach that could have been used is kd-trees. It equals LSH in speed with the current number of dimensions and has no margin of error. It is however possible to get a very precise approximation with LSH if appropriate values $l$ and $k$ are chosen [4].
4 Results

This section is dedicated to presenting the results from the study.

4.1 Troll data

The different parameters extracted from the trolls twitter feed are presented in the table below. The data from both sentiment tools are presented. Sentiment tool 1 refers to the method of letting the longest sentence in a tweet represent the overall sentiment score of that tweet. Sentiment tool 2 refers to the method of taking the average sentiment of all sentences in a tweet and letting that be the total score for the tweet.

<table>
<thead>
<tr>
<th>Troll name</th>
<th>Tweet frequency</th>
<th>Average time</th>
<th>Average sentiment</th>
<th>Deviation in sentiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>5.7500</td>
<td>745</td>
<td>1.6957</td>
<td>0.9526</td>
</tr>
<tr>
<td>T2</td>
<td>6.7500</td>
<td>638</td>
<td>1.8519</td>
<td>0.7554</td>
</tr>
<tr>
<td>T3</td>
<td>6.2500</td>
<td>745</td>
<td>1.9600</td>
<td>0.7736</td>
</tr>
<tr>
<td>T4</td>
<td>6.0000</td>
<td>758</td>
<td>1.9583</td>
<td>0.7895</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Troll name</th>
<th>Tweet frequency</th>
<th>Average time</th>
<th>Average sentiment</th>
<th>Deviation in sentiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>5.7500</td>
<td>745</td>
<td>1.8551</td>
<td>0.7587</td>
</tr>
<tr>
<td>T2</td>
<td>6.7500</td>
<td>638</td>
<td>1.9722</td>
<td>0.6240</td>
</tr>
<tr>
<td>T3</td>
<td>6.2500</td>
<td>745</td>
<td>1.9400</td>
<td>0.5713</td>
</tr>
<tr>
<td>T4</td>
<td>6.0000</td>
<td>758</td>
<td>2.0694</td>
<td>0.7992</td>
</tr>
</tbody>
</table>

The frequency and average time of tweets is of course the same with both sentiment tools.
4.2 Troll relative closeness

The following table was obtained by using the dimensions from troll #1-4 as query points. The numbers represent how many users are between the points representing the trolls. This is an indication of how close the trolls are to each other relative to other users. The distance is calculated with the euclidean metric.

<table>
<thead>
<tr>
<th></th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>*</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>T2</td>
<td>12</td>
<td>*</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>T3</td>
<td>6</td>
<td>5</td>
<td>*</td>
<td>1</td>
</tr>
<tr>
<td>T4</td>
<td>4</td>
<td>7</td>
<td>0</td>
<td>*</td>
</tr>
</tbody>
</table>

Notice that T3 and T4 are very close together with both sentiment tools. T2 is far off from the rest in both cases.

4.3 Comparison trolls and normal users

The following plots are 2-dimensional representations of the users values. Blue dots are normal users and red dots are trolls.
Figure 1: Average sentiment and standard deviation in sentiment using sentiment tool 1. All users are included in the plot.
Figure 2: Average sentiment and standard deviation in sentiment using sentiment tool 2. Three users with an average sentiment score under 1.6 were omitted from the plot.
Figure 3: Average time and amount of tweets per day. Extreme values omitted.
Users that average over 14 tweets per day are not showed in figure 3 as they are not very relevant to the study. A total of 10 users are outside the scope of the plot.

Worth noting is that three of the trolls are relatively close to each other in both graphs. Average number of tweets per day and sentiment deviation seems to be the parameters where the trolls are most similar.
5 Discussion

In this section a discussion is held concerning the method, result and other things related to the study.

5.1 Results

The data gathered from this study was not enough to find a troll farm. Sentiment tool 1 seems to have distinguished the trolls more than sentiment tool 2. Even with tool 1 the relation between the trolls relative to other twitter users were rather weak, as shown in table 3. Only troll 3 and 4 where each others nearest neighbour. Troll 1 and especially Troll 2 didn’t have a very strong relation to the other trolls. The abnormal parameter of troll number 2 was the average time, being much earlier in the day than the other trolls. This is possibly something that could have been prevented with stricter rules for tweet publications. Another observation that we can make from Figure 1 is that the trolls does not differentiate themselves very strongly from the normal users in the sentiment aspect. They do seem to have a slightly higher deviation in sentiment than the normal users. It was however not strong enough to single-handedly distinguish them. The trolls standard deviation in sentiment was surprisingly low considering that they were instructed to tweet opinionated and all had a relative neutral average sentiment score. This means that some of the opinionated tweets from the simulated trolls were classified as neutral by the sentiment tools, which indicates the tool had trouble with identifying the sentiment of the tweets. Whether this problem was caused by unclear tweets from the users or poor sentiment analysis is hard to determine, it is most likely a bit of both.
5.2 Method

The troll simulation was not ideal and we can attribute most of that to a lack of resources. If the participants were on the study full time a more accurate simulation would almost be guaranteed. It would also generate more data for the study. However there are a few things that could have been done to possibly improve the quality of the study without improved resources. The rules could have been stricter to make sure all trolls did the same thing. Since they have no training and lacking experience in the field the risk of them deviating from the original task is more present. Although three of the trolls were similar in their behaviour patterns, one went a different route and did not end up close to the others. Another thing that could be done to possibly improve results is to make the trolls sit in the same room. This is obviously closer to how a real troll farm works and the trolls may affect each other in some way.
6 Conclusion

It is not possible to detect the troll farm in this study using sentiment analysis and nearest neighbour search. With another data set or different tools the result could be different. More resources may have yielded a better foundation for the study and resulted in more analytic conclusions. This study can serve as reference when structuring future experiments in the same field.
REFERENCES

References


[10] Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing
REFERENCES


