Propagation Patterns of News on Twitter
- A Study in How News Propagate Through Twitter Via the Use of Bitly Links.

Spridningsmönster hos nyheter på Twitter
- En studie i hur nyheter propagerar på Twitter via Bitly-länkar.

Linnea Lundström
Sebastian Ragnarsson

Supervisor : Niklas Carlsson
Examiner : Marcus Bendtsen
Upphovsrätt


Copyright

The publishers will keep this document online on the Internet – or its possible replacement – for a period of 25 years starting from the date of publication barring exceptional circumstances. The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.

Linnea Lundström
© Sebastian Ragnarsson
Students in the 5 year Information Technology program complete a semester-long software development project during their sixth semester (third year). The project is completed in mid-sized groups, and the students implement a mobile application intended to be used in a multi-actor setting, currently a search and rescue scenario. In parallel they study several topics relevant to the technical and ethical considerations in the project. The project culminates by demonstrating a working product and a written report documenting the results of the practical development process including requirements elicitation. During the final stage of the semester, students create small groups and specialise in one topic, resulting in a bachelor thesis. The current report represents the results obtained during this specialisation work. Hence, the thesis should be viewed as part of a larger body of work required to pass the semester, including the conditions and requirements for a bachelor thesis.
Abstract

As so called fake news spread widely on the internet it is important to examine how they are spread, and thereby, how much of a problem they are. This thesis investigates how the spread of news articles on Twitter can be represented via a tree structure, as well as whether or not the trees have patterns that correlate to attributes such as the source of the shared news article and how many followers the original tweeter has. As part of the study a tool was built in Python 2.7 that, amongst other things, allows tracking and reconstruction of a news article’s propagation on Twitter.

It could be concluded that most links that are shared on Twitter propagate over a period of a few days and most retweets are made within the first twelve hours. We observe patterns suggesting that having more followers correlates to getting more retweets. Users who have few followers have to rely on their tweets being retweeted in a longer chain of users for it to reach a larger audience. Tweets that have a substantial spread often spread widely, but not especially deep. Finally, our results suggest that both the news site that created the article and the content of the article has an impact on how much it is retweeted.
Acknowledgements

We would like to thank our supervisor Niklas Carlsson for sharing his knowledge and guiding us past obstacles.

We would like to thank Olaf Nilsson and Filip Polbratt for sharing their code on retrieving tweets that we used. Also a special thanks to Olaf Nilsson for meeting up with us to discuss their code and their report more in depth.

We would like to thank Jesper Holmström and Daniel Jonsson for the open dialogue we kept during the extent of this study and for sharing their data sets with us.

We would like to thank Eric Lindskog and Jesper Wrang for reviewing our work and giving us useful feedback, which has allowed us to further improve our thesis.

We would also like to thank Daniel Roos for allowing us to use his computer server in order to run our data collection.

Lastly we would like to thank our friends and families who allowed us use of their phone numbers in order to register the Twitter accounts used for our tree accuracy evaluation.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abstract</strong></td>
<td>iii</td>
</tr>
<tr>
<td><strong>Acknowledgments</strong></td>
<td>v</td>
</tr>
<tr>
<td><strong>Contents</strong></td>
<td>vi</td>
</tr>
<tr>
<td><strong>List of Figures</strong></td>
<td>vii</td>
</tr>
<tr>
<td><strong>List of Tables</strong></td>
<td>viii</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Aim</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Research Questions</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Delimitations</td>
<td>2</td>
</tr>
<tr>
<td><strong>2 Theory</strong></td>
<td>3</td>
</tr>
<tr>
<td>2.1 Twitter</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Related Work</td>
<td>4</td>
</tr>
<tr>
<td><strong>3 Method</strong></td>
<td>7</td>
</tr>
<tr>
<td>3.1 Data Collection</td>
<td>7</td>
</tr>
<tr>
<td>3.2 Data Sorting</td>
<td>7</td>
</tr>
<tr>
<td>3.3 Creating Tree Structures</td>
<td>8</td>
</tr>
<tr>
<td>3.4 Evaluation of Propagation Tree Accuracy</td>
<td>11</td>
</tr>
<tr>
<td><strong>4 Results</strong></td>
<td>13</td>
</tr>
<tr>
<td>4.1 Correlation Between Follower Count and Retweet Count</td>
<td>13</td>
</tr>
<tr>
<td>4.2 Increase of Depth Over Time</td>
<td>15</td>
</tr>
<tr>
<td>4.3 Retweets over Depth and Time</td>
<td>18</td>
</tr>
<tr>
<td>4.4 Content Based Patterns</td>
<td>19</td>
</tr>
<tr>
<td>4.5 Correlation Between Tree Size, Breadth and Depth</td>
<td>20</td>
</tr>
<tr>
<td>4.6 Breadth and Depth Depending on Follower Count</td>
<td>21</td>
</tr>
<tr>
<td><strong>5 Discussion</strong></td>
<td>23</td>
</tr>
<tr>
<td>5.1 Results</td>
<td>23</td>
</tr>
<tr>
<td>5.2 Method</td>
<td>26</td>
</tr>
<tr>
<td>5.3 The Work In a Wider Context</td>
<td>27</td>
</tr>
<tr>
<td><strong>6 Conclusions</strong></td>
<td>28</td>
</tr>
<tr>
<td><strong>Bibliography</strong></td>
<td>29</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Example of a minimized tweet object.</td>
<td>8</td>
</tr>
<tr>
<td>3.2</td>
<td>Pseudocode for the algorithm used for attaching nodes in the tree.</td>
<td>10</td>
</tr>
<tr>
<td>3.3</td>
<td>Desired spreading pattern for scenarios including all types of posts.</td>
<td>11</td>
</tr>
<tr>
<td>3.4</td>
<td>Desired spreading pattern for scenarios including retweets and missing data.</td>
<td>12</td>
</tr>
<tr>
<td>4.1</td>
<td>A scatter plot showing an original tweeter’s follower count in correlation to how many retweets they got.</td>
<td>14</td>
</tr>
<tr>
<td>4.2</td>
<td>A box plot showing the means and medians of retweet counts for different intervals of follower counts.</td>
<td>15</td>
</tr>
<tr>
<td>4.3</td>
<td>A timeline with data points showing when a new level of depth was reached for all tweets in the three day set.</td>
<td>15</td>
</tr>
<tr>
<td>4.4</td>
<td>A timeline showing when new levels of depth are reached in the three day set, not counting roots.</td>
<td>16</td>
</tr>
<tr>
<td>4.5</td>
<td>Time line with data points showing when a new depth was reached for all tweets in the top set.</td>
<td>17</td>
</tr>
<tr>
<td>4.6</td>
<td>Time line with data points showing when a new level of depth was reached for all tweets in the random set.</td>
<td>17</td>
</tr>
<tr>
<td>4.7</td>
<td>Time line with data points showing when a new level of depth was reached for all tweets in the bottom set.</td>
<td>18</td>
</tr>
<tr>
<td>4.8</td>
<td>CDF of retweets that have been made when a depth is reached.</td>
<td>19</td>
</tr>
<tr>
<td>4.9</td>
<td>Percentage of retweets that have been made at different times.</td>
<td>19</td>
</tr>
<tr>
<td>4.10</td>
<td>Boxplot showing spreading patterns for news articles from different sources.</td>
<td>20</td>
</tr>
<tr>
<td>4.11</td>
<td>Correlation between tree size, breadth and depth.</td>
<td>21</td>
</tr>
<tr>
<td>4.12</td>
<td>Fractions of links whose propagation trees reached a certain depth.</td>
<td>21</td>
</tr>
<tr>
<td>4.13</td>
<td>The relation between breadth and depth in a propagation subtree and how many followers the original tweeter has.</td>
<td>22</td>
</tr>
<tr>
<td>4.14</td>
<td>Medium values for the relations between breadth and depth for intervals of follower counts, together with every data point in the set on a logarithmic scale.</td>
<td>22</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Statistics on how much data that is missing in the different sets.</td>
<td>13</td>
</tr>
<tr>
<td>4.2</td>
<td>Retweet rates for different news sources, sorted by data set.</td>
<td>20</td>
</tr>
</tbody>
</table>
1 Introduction

As it has become easier to access news from a variety of online publishers, the classic printed newspaper is being abandoned. This comes with many benefits but also its fair amount of drawbacks. On social media, news reach a broader audience every day and information about the world is abundant for many. However, this also raises concerns regarding authenticity and objectivity of the news presented. The easier and more frequent a task becomes, the less thought is put into it and it becomes more of an everyday chore, much like opening a door or brushing ones teeth. This shift in focus opens up for mistakes as information propagation is not as trivial as first might seem.

1.1 Motivation

During the time period of the 2016 US election, the term fake news was used commonly to describe information that was believed to differ from the truth. As it can be difficult for consumers to assess whether a news article is fake or not, fake news can spread just as easily as any other form of news [1]. This spread can also be amplified by the fact that fake news often have a more compelling and exaggerated content.

It has even been suggested that fake news played a role in getting the current US president, Donald Trump, elected. One reason for this is that fake news that favored him spread more widely than those favouring the opposing candidate Hillary Clinton. Studies have also shown that many believe the fake news they read to be true, and that fake news on the social medium Facebook were shared more than real news [1].

When discussing how the source of a news article affects how it spreads, studies have shown surprising results. One might believe that a user, on Twitter for example, would share news from outlets that share their ideology, but that is suggested to be false. As people share more news they seem to include sources from the opposing side of their viewpoint. Yet, it should not be ruled out that selective exposure plays a role in what gets shared [2].

In conclusion, proofreading and critiquing of sources is important and a skill that, unfortunately, few seem to exercise in the age of information. The key word here however is seem, as we do not know for certain. Before developing and deploying counter measures, it is important to know what you are countering in the first place. It is therefore both interesting and valuable to search for patterns in what gets more exposure, depending on whether it is true or false, politically biased, and what the source of information is.
1.2 Aim

In this study, we aim to find a methodology to be used for in-depth visualization and analysis of the propagation of news on social media. More precisely, we aim to build a tool for creating a tree structure model that visualizes how an article is spread through Twitter via quotes and retweets, two ways in which posts are shared on the site. Through identification of sharing patterns, the thesis aims to characterize differences in how tweets are shared depending on the article included in them, as well as other traits of the tweets. To enable investigation of how fake and real news and spread, the study sets out to develop a tool that makes such research possible at a larger scale. Ideally, the tool should also be adaptable and help retrieve information about individual tweets.

1.3 Research Questions

1. How should influence on Twitter be assessed in order to properly connect tweets and retweets regarding a specific news article in a tree structure?

2. Based on the sources of news articles included in tweets, what differences are there in terms of their diffusion?

3. What other patterns can be found when comparing different tree structures, depending on for instance the amount of followers that the original tweeter has?

1.4 Delimitations

We have chosen to limit the tweets we collect to only those containing Bitly links. We will only consider retweets that were made using the official retweet functionality of Twitter, and ignore the unofficial "RT @user" tag, which is sometimes added to a tweet to signify that another user’s tweet is being quoted.
2 Theory

A first step in studying how data about tweets can be visualized in a tree structure is examining exactly what data that can be acquired. In other words, we need to look into the possibilities and limitations of the Twitter API. It is also wise to delve into previous research to see what realizations and obstacles others have encountered when studying similar subject matters.

2.1 Twitter

Since it launched in 2006, Twitter has become one of the biggest social networking sites in the world. It allows users to share their thoughts in short messages limited to 280 characters called tweets. Tweets are shared amongst users through the network via connections established by who they follow and who in turn follow them. Users consist of people as well as organizations and news outlets.

A study titled “Why we twitter: understanding microblogging usage and communities” [3], investigated the main usages of the microblogging site. It was concluded that most users use it to talk about events in their daily lives. Many hold conversations on the social medium by responding to each other’s tweets. Besides that, it is popular to share information or links, mostly using URL shorteners due to the character limits set by Twitter. Lastly, users share news or comment on current events.

The Twitter API

A post on Twitter can be one of three types: an original tweet, a retweet or a quote [4]. An original tweet is content created directly by the user, in our case containing a link to a news article that the user has supposedly read and shared to their followers. A retweet is a post that could be said to echo another post as it links back to it. These are generally posted by other users than the one who posted the original. A quote is a retweet with a comment added by the quoting user. The Twitter API includes methods that can be used to find out if a post is an original or if it is a quote. In the case that it is a quote, information is also presented regarding which tweet was quoted. This information however is not included when it comes to retweets, then we can only know what the original tweet was. Therefore, if a post is a retweet of a retweet, you cannot know who the intermediary retweeter was.
Also important to consider when processing data provided by the API is a phenomenon that Kupavskii et al. [5] found during their research. They state that when a post is retweeted by two or more people that a user follows, the user will only see the post from the first retweeter in their home timeline. We also confirmed this by creating four Twitter accounts, where two of them followed an account that posted a tweet. They both retweeted the original and the fourth account following these two would only have one of the retweets appear in its home timeline. The visible retweet was the one that was chronologically first. Through testing we saw that if the user had also followed the original tweeter only the original tweet would be visible in the users home timeline, and it would not be apparent that a friend of theirs had retweeted it.

Stream Processing and Batch Processing

There are two main ways of gathering data from Twitter, one being stream processing and the other being batch processing. Stream processing, in terms of the Twitter API, is the act of downloading JSON formatted data containing tweet objects as they appear on the social medium. This can be used in order to collect tweets as soon as they appear and scan them for Bitly links.

Batch processing can be used in order to collect specific data that has been stored for an undefined amount of time, for example to get information about a user. Utilizing the API in such a way allows us to gather information about who a certain user follows. This information can then be used to determine which tweet the user saw in their home timeline and thus retweeted. This is however limited to 15 requests over 15 minutes, where each request can return up to 5000 friends. The same can be done with followers [6].

2.2 Related Work

In recent years, analyzing how information is spread on social media has become more relevant. There is therefore plenty of research available on how information on Twitter spreads. Especially when investigating factors that make a news article get retweeted, and how the amount of followers a user has impacts their influence, there is more than enough research available.

The Dynamics of Propagation on the Internet

Compared to traditional media, the internet has given news stories longer reach and a much faster way of going across the world. Many studies have been conducted in order to find out more about these complex spreading patterns and how they propagate. One study conducted by Leskovec et al. [7] investigates how phrases and words are used on the internet and how they propagate. They investigate phrases from news articles and from social trends, so called memes. They write about how phrases mutate due to being shared and other affecting factors. By bringing the mutated phrases back to their original meaning they can track usage of that phrase and make observations based on it. They conclude that there is a time period of 2.5 hours between the peaks of attention to a phrase in the news media and in blogs.

De Leng et al. [8] have studied the dynamics of propagation on the internet but with focus on major sporting events and how news related to the event propagate on Twitter. They investigate and analyze the real-time nature of Twitter and how news can spread even faster during these sorts of events. They conclude that Twitter experiences real-time spikes in usage as goals and other highlights occur in games. Alongside they also observe an increase in usage during the period after a game ends.
Counting Clicks and Classifying Twitter Posts

In a Bachelor thesis titled “Counting the Clicks on Twitter” [9], Polbratt and Nilsson state their primary goal to develop a tool that will crawl Twitter and find out what content is being read. In their study they used the streaming process available through the Twitter API in order to fetch tweets. Alongside this they also investigated if there is any difference in how biased and non-biased content is read on the site. In classifying articles they used two main methods. The first being a machine learning tool that gave inconclusive results. Secondly they manually classified sites in order to assign them with a probability of how likely they were to share content of certain classes. Using these two methods combined resulted in a somewhat accurate classification metric. They concluded that their study was not extensive, but what data they had gathered and analyzed pointed towards there being a difference in how links are clicked based on their classification.

Predicting Propagation Depending on User and URL

Bakshy et al. [10] construct and analyze what they refer to as Twitter cascades, which is essentially a tree structure. They come to the conclusion that the biggest spread of information comes from a large target audience. In other words, if a user has many followers the odds are simply higher that the information will be passed on by some of them. They also write about what they refer to as moderately sized cascades, cascades of retweeting that are very large and propagate very quickly. They explain that most of their observations were made in regards to these events and that they occur extremely rarely. They also conclude that tweets that include links tend to be retweeted more. Boyd et al. [11] make similar observations. In their study, they have collected random samples of tweets and retweets respectively and in the samples, 22% of tweets had a URL in them, whereas the number was 52% for retweets.

Suh et al. [12] have analyzed the features of 74 million tweets with the purpose of determining what factors affect retweetability. They use retweet rate as a measurement of how much a tweet is shared. Here, the retweet rate is calculated by dividing the number of retweets by the number of original tweets. They concluded that factors such as which URL:s were used, how high the user’s follower count was and how old their account was increased retweetability. They also concluded that how much a user tweets does not have the same impact.

Kwak et al. [13] investigate 106 million tweets and the users they were made by. They rank the top 20 Twitter users in regards to how many followers they have and the popularity of their tweets. Results show that the overlap of users in the two rankings is small, suggesting that having many followers does not correlate with how many retweets you get. Of the users that do overlap, two of them are CNN Breaking News and The New York Times, which implies that their followers more often find their tweets worth spreading. This seems to be the case for other news media as well, as users such as The Huffington Post, ESPN Sports News and NPR News are in the top 20 for number of retweets, but not for number of followers. The order of how much they are retweeted goes CNN at rank 5, New York Times at rank 8, and The Huffington post at rank 16.

Cha et al. [14] investigate three different measures of user influence. The first is indegree, meaning how many followers a user has, the second is retweets, and the third is mentions. Their study showed that although indegree is a measurement of a user’s popularity, it does not necessarily correlate with how many retweets their tweets get. It is primarily the value of the content of a tweet that determines how much it gets retweeted, and the value of the name of a user that determines how many mentions they get.

Correlation Between Falseness and Propagation

In a study conducted by Vosoughi et al. [15], tweets about news stories from 2006 to 2017 involving three million people were investigated in terms of diffusion depending on their
classification. The articles were classified as true or false with the help of fact-checking organizations. They investigated how long it took for true and false news respectively to reach 1500 people. A cascade was made to show how content was retweeted. The results were that it took six times longer for true news than it did for false news. It also took 20 times longer for true news to reach a cascade depth of ten. Differences could also be shown depending on the type of false news an article was about. The ones with political content spread faster, deeper and wider. The study also showed that false news were 70 percent more likely to be retweeted than true news.

Methods For Constructing a Tree Structure

One way of finding out how a tweet has been shared, in order to construct trees that visualize the propagation, is to use the method of Yong and Counts [16]. They used the built in mentioning feature of Twitter to collect data on which users followed which other users. They made a connection between tweets by gathering instances when users used the @username tag, followed by a subject or a link that the tagged user had previously tweeted about. They could from this conclude that it was a reference to an earlier tweet the user made and connect the two.

Another approach is discussed by S. Goel et al. [17], who describe a method for constructing a tree structure, a diffusion tree, for tweets that contain a specific piece of content, for example a URL. Each node represents a user and edges link them to their parent. To decide on a parent they first identify a set of potential parents, which is every user that could have shared the content and have it appear in this user’s timeline. What appears in a user’s timeline is content posted by friends of that user, and tweets that were officially retweeted by those friends. They state that a tweet only appears in a user’s timeline once, even if it was retweeted by multiple friends.
To conduct this study some essential steps needed to be carried out. Firstly, a set of data containing tweets with Bitly links was needed. Bitly links were chosen as it is a mean of keeping track of when the links are clicked. They are also used broadly on Twitter in order not to reach the character limit as fast. Secondly, there was a need for a script that could build tree structures from said data. Included in this is also making an algorithm for making the retweet chain as accurate as possible, while also minimizing the time consumption for creating it.

3.1 Data Collection

The data that was used in this study was extracted from four different data sets. For collecting the initial set, the code written by Polbratt and Nilsson [9] was used. Using their tool gave us the opportunity to focus more heavily on pattern recognition and analysis of said patterns instead of on collecting data. Their script used Twitter’s streaming API to collect data regarding posts on Twitter that related to Bitly links. The tweets were collected over a period of slightly more than three days. It was saved in text files where each post was structured in a JSON format. The data set will hereafter be referred to as the three day set.

The three remaining sets were collected by another group of Bachelor students. They proceeded from the same code we used in our collection for the three day set and built additional features on top of the existing ones. All their sets of tweets were collected over the same time period of twelve days. One of them will be referred to as the top set, and consists of tweets containing links that have been clicked at least 450 times during the first twelve hours of the collection process. The second one will be referred to as the random set, and consists of a random selection of tweets containing about 25% of all links. The last one contains the remaining links and associated tweets, and will be referred two as the bottom set. Note that there is originally overlapping data in these three sets, but such data is excluded in this study.

3.2 Data Sorting

The data collected contained tweets in the form of JSON objects, sorted only by the time they were retrieved. This resulted in large amounts of data where objects that related to the same
3.3 Creating Tree Structures

Creating a tree structure, a sort of propagation tree, the Python module `anytree` was used. `anytree` contains a class called `AnyNode` that is customizable in that you can choose any attribute for any node, which is preferable in this case as it can vary what attributes of a post that are of interest. By setting a value in the `parent` attribute, you can attach a node to another and thereby make it its child.

When rendering a tree with `AnyNode`, a specific node must be given as input to be used as a root. This means that only the nodes that are descendants of this node will be visible in the tree. However, in this study we will want to include multiple roots if the same link has been tweeted by multiple users. A work-around to this problem is to create a new Python class.
that acts as a tree class, and iterates through all nodes that do not have parents and renders multiple trees with them as roots. This tree class also retrieves information about a specific node from our collected data. This can be done by knowing its ID in the tree structure or the ID of the Twitter post the node is representing.

In many cases, linking a node to its parent is as simple as adding the parent node as an attribute when creating the child node. In other cases, there is no node with the ID of the parent. This is due to the fact that not all tweets are captured during the data collection. A solution to this is creating an artificial parent node to assign to the child node. This is done by utilizing the fact that all JSON objects of retweets and quotes contain a reference to the original post with all of its attributes.

Handling Different Types of Posts

The Twitter posts of this study are of three different types; original tweets, retweets and quote\(^1\). Original tweets do not have a parent and therefore, they are roots. This means that there will be one root for each user that made a new tweet about a specific URL.

The process of linking a retweet to the tweet it references is no longer as convoluted as it was in 2010 and Yong and Counts \cite{16} conducted their study. The twitter API has been extended to include more information about these relations. Although we will not use their method in order to connect tweets, it should be noted that it can be used in order to ensure the accuracy of our connections. Even though there is more support available in the API now, there are still some uncertainties that remain. One being that we still cannot know for certain which user influenced the retweeting user in such a way that they would in fact make a retweet. Much like Yong and Counts \cite{15} we assume that a fair simplification is to assume that if a user A retweets some content, and A follows an original tweeter B, it was B that influenced A to retweet. If A does not follow B, the first person that A follows who retweeted this post is probably the person that exposed A to the content and made A retweet it. This argument can be made recursively to cover any levels of the tree. If no user that A follows has tweeted or retweeted the content, credit is given to the original tweeter and a connection is made.

Extracting Friends and Followers of a User

To be able to make connections as described above, one of three approaches can be used. The first is to extract friends for each user that is not an original tweeter in order to find who should be their parent. The second is to extract followers for users in an attempt to be able to attach their children immediately when they are found. A third approach is to use a combination of the two, which is what was done in this study. This gives the possibility of improving the time consumption of building trees, if used correctly.

The extracting process utilizes functionality within the Twitter API to collect data on a user’s friends. By feeding it a user it returns a list containing that user’s entire catalogue of friends. The request limit of 5000 friends per request and sixty requests per hour is something that proved to be a problem as most of our trees contain well above fifteen posts. This is solved by maximizing the amount of requests that can be made within the time limit, followed by waiting until the limit is reset and more requests can be completed. This is a lengthy process caused by the rules set by the Twitter API, but in the end it allows for a more accurate propagation tree. This is done using batch processing and so the data is not retrieved in real time, it is instead available at any time.

The algorithm used to minimize time consumption is as shown in Figure \ref{fig:batch-processing}. First, all posts regarding the link are iterated through. A dictionary is built containing each user found in a retweeted_status or a quoted_status together with the amount of times

\(^1\)As quotes are a form of retweets, they will from now on be included in that term.
they appear. After that, posts are iterated through again and individually attached to the propagation tree. Each time an original tweet is found, it is assessed whether it should save time if followers were requested for that user. This is done with help of the dictionary where you can look up the maximum number of children that the node might have. If the estimated time for requesting their friends is higher than the time it would take to request this tweeter’s followers, the second should be done. Important to note, however, is that there is no guarantee that the potential children will appear in the follower list of this tweeter, in which case their friends will have to be requested anyway.

```python
def getRepostedUsers():
    for post in posts:
        if post is quote or retweet:
            -> get original tweeter id
            -> search for original tweeter in dictionary
            -> update number of occurances

def makeTree():
    for post in posts:
        if original tweeter:
            if tweeter in dictionary:
                -> occurances = dictionary[tweeter]
                if occurances > tweeter[followerCount]/5000:
                    -> get followers
            else:
                -> get original tweeter
                -> search for user in follower list
                if user found:
                    -> attach to original
                else:
                    -> get friends
                    -> search for friend that reposted
                    if friend has reposted:
                        -> attach to friend
                    else:
                        -> attach to original
```

Figure 3.2: Pseudocode for the algorithm used for attaching nodes in the tree.

**Extracting General Tree Data**

As more and more trees are created, it becomes more difficult to get an overview of what data to further investigate. A way to solve this is to gather data about each tree as a whole. Information that can be interesting to save is the size of the tree, its depth, its breadth and the time span over which the tweets in it were made. The depth of the tree is the longest chain from root to leaf, whereas breadth in this case is the maximum number of people who retweeted from the same user.

AnyNode has a built in attribute that shows the height of a node, which is the same as the highest amount of levels in a chain from said node to a leaf node. This can be used to find the depth of the propagation tree if done for all root nodes.
3.4 Evaluation of Propagation Tree Accuracy

The breadth can be found with a recursive solution, where the number of children of a node is calculated and concurrently, the number of children for every child found, is calculated.

3.4 Evaluation of Propagation Tree Accuracy

To be more certain that propagation trees were built correctly, the algorithm was run on a set of evaluation data. The evaluation data was gathered by using ten Twitter accounts set to tweet, retweet and quote each other in accordance to a predetermined plan. With this approach, any desired spreading pattern could be specified beforehand. The desired tree could then be compared to the tree constructed by the algorithm to determine accuracy.

Testing All Types of Posts

The evaluation data was used to test that the trees were built correctly for the following scenarios:

- Multiple users posting original statuses with the same link
- A user quoting something they saw only because someone they follow retweeted it
- A user retweeting a quote and not the original tweet

A chain of posting, quoting and retweeting that would test all of the above scenarios is shown in Figure 3.3. To get this tree, Q4 and R4 must not follow O2, as that should result in them being connected to O2. If Q2 is posted after Q1, Q2 must not follow Q1.

As it turned out, there is no simple way to collect quotes when having specified users to follow. Even as this study concluded it was still unclear to why this was, as earlier data collections included quotes. Several other methods for retrieving this data were tested, such as tracking the specific Bitly link or a phrase that was included in all quotes. The issue with these approaches was that the streaming API does not collect all tweets, and if specific users are not followed, they are easy to miss. The test was therefore altered to handle only retweets and original tweets.
To still be somewhat certain that quotes were handled correctly, a JSON object from the data collection that was guaranteed to be a quote was compared to an object that was a retweet. It could be observed that they were very similar in what data was included. The differences that existed should not affect how the two types were handled by the algorithm. The important difference between the two is that a quote has a quoted_status attribute and a retweet has a retweeted_status attribute containing the quoted or retweeted post. In the case that a retweet is of a quote, it will also have a quoted_status attribute, but it will be nestled inside the retweeted_status and is therefore not reachable with the same Python command.

**Testing Retweets and Missing Data**

The new test was one that should result in propagation trees as shown in Figure 3.4. With this test, the following scenarios are evaluated:

- Multiple users posting original tweets with the same link (O1, O2, O3)
- Users retweeting posts made visible to them by a retweet from a user they follow (R1, R2, R3)
- An original tweet not being in the data collection (O2)
- A user retweeting a post made by someone they do not follow (R4)
- A user retweeting a post made by someone they follow that was also retweeted by someone they follow (R5)
- A user retweeting something they saw only because someone they follow retweeted it (R6)
- A user retweeting a post that was retweeted by two users that they follow (R7)
- A user posting a link that was posted before, but with a text added (O3)

![Desired spreading pattern for scenarios including retweets and missing data.](image-url)
Results

As data from the top, random, and bottom set was more recent and had been collected over a longer period of time compared to the three day set, a decision was made to mainly focus on tweets from those in the analysis of propagation trees. In Table 4.1 statistics on how many retweets that are missing from the different sets are shown. It can be observed that the top set and the random set contain a larger fraction of all tweets that have been made regarding the links in the sets, whereas more data is missing in the bottom and three day set.

<table>
<thead>
<tr>
<th>Set</th>
<th>Nr of links</th>
<th>Nr of tweets</th>
<th>Found retweets</th>
<th>Total retweets</th>
<th>Missing (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top</td>
<td>43</td>
<td>6879</td>
<td>5654</td>
<td>8351</td>
<td>32,30</td>
</tr>
<tr>
<td>Random</td>
<td>141</td>
<td>1940</td>
<td>1246</td>
<td>2040</td>
<td>38,92</td>
</tr>
<tr>
<td>Bottom</td>
<td>1345</td>
<td>11537</td>
<td>6569</td>
<td>164176</td>
<td>96,00</td>
</tr>
<tr>
<td>Three day</td>
<td>263</td>
<td>46470</td>
<td>16973</td>
<td>428067</td>
<td>96,03</td>
</tr>
</tbody>
</table>

Table 4.1: Statistics on how much data that is missing in the different sets.

4.1 Correlation Between Follower Count and Retweet Count

A scatter plot, as seen in Figure 4.1, was used in order to illustrate any possible correlation between follower count and retweet count. Each point in the plot represents a tweet, retweet or quote posted by a user and collected in one of our three data sets. The size of each data point represents the size of the propagation tree it belongs to. Equally sized trees are also the same color in order to increase the readability of the graph.

It can be seen how an increase in followers only sometimes correlates to an increase in retweets. What this means is that a large follower base does not necessarily mean a large amount of retweets. It seems however that it helps as a larger amount of retweets can be seen when looking at bigger follower bases, which is to be expected. What cannot be seen however is a correlation between this trend and tree size. By observing how data points of similar sizes are spread out across the graph it can be concluded that users who share the same link do not show any specific shared traits when it comes to followers and the amount of times their tweets are retweeted.
4.1. Correlation Between Follower Count and Retweet Count

The means and medians of Figure 4.1 can be illustrated using the box plot shown in Figure 4.2. Although there were outliers they are not visualized in this graph as they are few in comparison and they decrease the readability of the graph tremendously. It should however be noted that this means that our data sets have tweets with great variation in how much they are retweeted. The graph is separated into five groups of four boxes, one box for each of our four data sets, with each group representing followers grouped by powers of ten from Figure 4.1. Data points from 0 to 10 followers were excluded as there were not a sufficient amount of them to produce a fair representation.

In Figure 4.2 the increase of retweets can be observed very easily. It can be seen in the first three groups, in between 10 and 10 000 followers, that the vast majority of tweets only receive one to three retweets. This is increased to in between one and ten tweets in group four and finally in group five it lands at about one to a hundred retweets. It can also be seen in group five that the three day set and the bottom set still both contain a lot of tweets with only one or two retweets. When observing the random set and top set it can be seen that they both have significantly fewer tweets with such a low retweet rate.

Figure 4.1: A scatter plot showing an original tweeter’s follower count in correlation to how many retweets they got.
4.2 Increase of Depth Over Time

As earlier stated, information about the general properties of propagation trees has been extracted during the process of creating the trees in question. Among this data are time stamps for when each tree gains depth. Figure 4.3 until Figure 4.6 illustrate when these depths are reached for tweets in the different data sets individually. Figure 4.4 illustrates only retweets and quotes in the three day set, in order to get a more detailed view.

Figure 4.2: A box plot showing the means and medians of retweet counts for different intervals of follower counts.

Figure 4.3: A timeline with data points showing when a new level of depth was reached for all tweets in the three day set.
4.2. Increase of Depth Over Time

Figure 4.4: A timeline showing when new levels of depth are reached in the three day set, not counting roots.

Upon inspecting Figure 4.3, some observations can be made. The time span of the graph is over a much longer period than the three days of the data collection. In some cases the graph is uneventful for a few days and then suddenly gains multiple levels over hours, or even minutes. Tweets in this set do not seem to reach deep very often, with as few as one having more than four levels. While most propagation trees have their original tweet included in the data set, some of them do not. In Figure 4.4 the same data is represented, but all original tweets are excluded, which should make it more clear what happened during the days of only collecting. It can be observed that the somewhat deeper propagation trees reached their full depth over the first 36 hours, whereas shallow trees took longer.

In Figure 4.6 no apparent patterns are seen as to if and when certain depths are reached. When observing the data for the top set in Figure 4.5 new depths tend to be reached faster than for the random set and they are also deeper in general. In Figure 4.7 similar patterns as for Figure 4.3 can be viewed. Only a fraction of tweets reach deep while the rest are shallow over a long time period. However, level two is often reached for trees in the bottom set, whereas many of the trees in the three day set only have one level.
4.2. Increase of Depth Over Time

Figure 4.5: Time line with data points showing when a new depth was reached for all tweets in the top set.

Figure 4.6: Time line with data points showing when a new level of depth was reached for all tweets in the random set.
4.3 Retweets over Depth and Time

Figure 4.7: Time line with data points showing when a new level of depth was reached for all tweets in the bottom set.

Figure 4.8 shows the probability that a randomly chosen retweet has been made when a certain depth is reached. Figure 4.9 illustrates what percentage of all retweets have been made at a certain time. Here it can be seen, that for the top set, it usually takes only six hours before half of all retweets are made after a tweet has been posted. For the random and bottom set, the number is almost double. For all four sets, it is true in most cases that tweets are no longer being retweeted after four and a half day.

When considering depth, Figure 4.8 shows that new levels are reached at a fast pace compared to the rate of retweeting.

In order to get as accurate data as possible, all links that had a propagation over a longer period than the length of the data collection in question have been excluded from these two graphs, as there is most likely data missing.
4.4 Content Based Patterns

Figure 4.8 illustrates retweet rates for articles from a given news source. Here, retweet rate is a measurement of how many retweets there are for each original tweet regarding a specific link. Patterns that exist can be overviewed easier using Table 4.2. Here, a medium value for the retweet rates of articles from the same news source is shown together with the number of links that were processed regarding this source. It can be observed that depending on the source of an article that is shared on Twitter, the rate of the spread differs. The medium rate is highest for articles from The Times, BBC and Fox News, whereas it is the lowest for articles from CNN and The Guardian. Figure 4.10 shows similarities between different news sources.
4.5 Correlation Between Tree Size, Breadth and Depth

in how the retweet rate varies. Variations, and medians, are similar for The Times and BBC, and for Huffington Post, CNN and The Guardian.

<table>
<thead>
<tr>
<th>News source</th>
<th>Medium rate</th>
<th>Nr of links</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Top</td>
</tr>
<tr>
<td>Washington Post</td>
<td>1.5</td>
<td>-</td>
</tr>
<tr>
<td>The Times</td>
<td>3.4</td>
<td>-</td>
</tr>
<tr>
<td>Huffington Post</td>
<td>1.2</td>
<td>-</td>
</tr>
<tr>
<td>CBS</td>
<td>0.5</td>
<td>-</td>
</tr>
<tr>
<td>CNN</td>
<td>0.9</td>
<td>-</td>
</tr>
<tr>
<td>Fox News</td>
<td>3.0</td>
<td>15.2</td>
</tr>
<tr>
<td>Breitbart</td>
<td>1.7</td>
<td>3.9</td>
</tr>
<tr>
<td>BBC</td>
<td>3.1</td>
<td>4.1</td>
</tr>
<tr>
<td>The Guardian</td>
<td>0.8</td>
<td>2.1</td>
</tr>
</tbody>
</table>

Table 4.2: Retweet rates for different news sources, sorted by data set.

Figure 4.10: Boxplot showing spreading patterns for news articles from different sources.

4.5 Correlation Between Tree Size, Breadth and Depth

Figure 4.11 shows the relation between breadth and depth in different propagation trees and different sets. The size of each point correlates with the amount of nodes in its tree. Some correlations can be seen, especially in that larger trees are broader. It can be seen that more trees propagate broadly than deeply, this as the amount of trees decrease tremendously in depth but not in breadth. This is strengthened as Figure 4.12 illustrates that only a small fraction of all trees have a depth of more than four and 50% of them have only one level. It can be seen that the top set is the set that brakes this trend the most. This as the top set has an unusually high amount of trees that propagate deeply, this even though it has relatively few data points.
4.6. Breadth and Depth Depending on Follower Count

Figure 4.11: Correlation between tree size, breadth and depth.

Figure 4.12: Fractions of links whose propagation trees reached a certain depth.

4.6 Breadth and Depth Depending on Follower Count

Figure 4.13 illustrates how the depth to breadth relationship of a propagation tree correlates with how many followers the original tweeter has. The relationship is calculated for each subtree, meaning that there is one entry for each original tweeter and the cascade connected to them specifically. In Figure 4.14, an average of these values for different intervals of follower amount can be seen to get a better overview. As can be seen here, there is a difference in how a tweet propagates depending on how many followers the original tweeter has. In cases where the original tweeter does not have many followers, the tweets tend to reach more depth compared to their breadth. Upon further inspection of Figure 4.13, it seems that even though
there is less data for larger follower counts, the data points that there are are more spread out the higher the follower count is, whereas for low follower counts they are more compact, or heavy, for lower values.

Figure 4.13: The relation between breadth and depth in a propagation subtree and how many followers the original tweeter has.

Figure 4.14: Medium values for the relations between breadth and depth for intervals of follower counts, together with every data point in the set on a logarithmic scale.
Using our chosen graph types we have made some observations in regards to building and evaluating our propagation trees. Alongside this we also discuss the social and ethical aspects of collecting and analyzing data in the way we have been for the extent of our study.

5.1 Results

The propagation trees that were created showed interesting, however not always unexpected, patterns in regards to how fast, deep, and broad, links are spread on Twitter, as well as how it has an impact who is making a tweet, and the source of the link in question. Other than this, what must also be discussed is differences that were seen depending on what data set a link was included in, and how that impacts its diffusion.

Three Day Set Evaluation

As could be observed regarding how trees in the three day set get deeper over time, many tweets did not seem to propagate much for days, and then suddenly multiple levels were gained fast. This pattern was seen for tweets that had a time period for more than three days. Here, a reasonable assumption is that a lot happened between the time of the first tweet and the time we started capturing. Important to emphasize it that for every graph reaching over a longer period than three and a half days, there must be data missing. Therefore, it can be concluded that three days is not a sufficient time period to collect data for this type of study. Looking at Figure 4.4 this becomes even clearer. The graphs that did not reach higher than level two might seem as if they would have if the time period was longer, since that level is often reached at the end of the period. We can hence assume that some of the data extracted from this set is misleading.

Comparison of Tweets in Sets

An interesting observation can be made when comparing how fast new depths are reached for the top set and the three day set respectively. For the three day set, we earlier concluded that the time period of the collection was insufficient. Yet, when looking at the top set, all trees have stopped gaining new levels after three days. This might be due to the fact that the
top set contains more popular links, which in turn could mean that their spread tapers off more rapidly as less time is required to reach out on Twitter. Another reason might be that since so much data is missing in the three day set, it looks as if new levels are reached when in fact they reached those levels days ago. To further strengthen this theory, one can look to the left in Figure 4.3. Here, it can be noted that the trees for which we captured the original tweets taper off quite fast, which suggests that they might not have gained more levels after the three day period. Although the bottom set did not share the problem of a short collection period with the three day set, it did also have a large amount of missing data. Therefore, the same assumption can be made that new levels are reached, but the tweets responsible for that are not included in the set.

For the random set it is also true that new levels are rarely reached after three days. For this set, there is more of a difference in how fast these levels are reached compared to in the top set. This is probably because the links included are in fact, randomly chosen. This means that they do not have as much in common as the top set.

The general conclusions that can be drawn are that the top set and the random set have the most reliable data, and patterns seen for tweets in these two sets can be trusted to a greater extent.

Patterns Depending on Tree Size

During our study it has become apparent that there is a difference in how many nodes each tree contains. Some contain only one while others can contain upwards thousands. This can most easily be observed in Figure 4.1 by looking at the size and coloring of the data points, in other words the size of its tree. Our results show that spreading patterns, in regards to follower count and retweet count ratio, are not affected by this variation in size.

When inspecting depth and breadth depending on propagation tree size there is, as earlier stated, more of a pattern. Propagation trees that are larger are wider compared to their depth. This is understandable as it takes more for a tweet to spread deeper, compared to it spreading wider. For it to spread deeply it has to be a user on the outermost level that influences someone to retweet, which means that the potential retweeter must not follow anyone else that retweeted before this user. With this in consideration it becomes apparent that it is much more likely that a new node will contribute to breadth rather than depth.

As shown in Figure 4.1 it can be seen that articles in the top set propagate unusually deep compared to their breadth. Our leading theory is that since the top set contains articles that are particularly interesting and popular we can assume that it panders to a wide audience. These articles have a much higher likelihood of propagating deep because of two reasons. The first being that if an article is popular it is more likely overall to be clicked. This means that the dissipating affect of fewer followers being reached in deeper levels is not as impactful as it would otherwise have been. The tapering off will instead happen at a later layer. This would however also mean that those trees would be unusually broad, something that we do not see. This leads us to believe that this is not the main reason. Instead we think social groups and their interests might be more relevant. If a user retweets something because they find it interesting, other people will probably do so because of the same reason. If an article panders to a broad audience this means that it is not only being spread between users with a certain shared niche. If, for example, an article is about a local football team winning a game, it will be popular amongst those who live in that town and enjoy football. The person tweeting and those retweeting probably follow each other as they are acquaintances and that tree will therefore be broad, not deep. If the subject matter is the winner of the world championship however, no such closed group will exist and the tweet can be shared outside of the original tweeters social group. Removing the limiting factor of social groups with shared interests will allow the tweet to propagate deeply and not necessarily broader as not all followers will have this shared interest.
5.1. Results

Retweets over Time and Depth
As could be seen in Figure 4.9, the rate at which retweets were made decreased over time and most of them were made over only a couple of hours. When looking at depth, Figure 4.8 showed that depths are reached at a quite fast pace compared to retweets being made. This means that even though users retweet and thereby make the tree deeper, other users are still mostly retweeting from users that tweeted earlier. This is compatible with our statement that it is easier to reach breadth, than it is to reach depth. There is simply a much higher likelihood that a user who posts a retweet follows someone that is not amongst the outermost nodes currently in the tree. In that way, the amount of nodes on a level of a tree increases considerably even after the next level has been reached. This alongside with what can be seen in Figure 4.9 we can conclude that new depths are reached under a short period of time and even though a new level has been reached in the tree a lot of activity still occur in the earlier levels.

Correlation Between Follower Count and Retweet Count
As Bakshy et al. mention in their paper [10], they believe there is a direct correlation between the amount of followers a user has and how many retweets the user’s tweets get regardless of their content. Going by that, we should see a trend in our data which suggests that users with more followers always have more retweets, but from our data we observed that this is not the case and the closest we get to matching their results is with users who have a follower count of above 100 000 and that are from the random set or the top set. This suggests that the theory presented by Bakshy et al. does not seem to apply to our data, as such a correlation would present itself as a distinct linear increase in averages but instead we only see a slight increase in the maximum values.

This might be explained however by that Bakshy et al. draw their conclusions based on what they call moderately sized cascades. It might be that during the total two weeks of accumulated collecting, done by us and the other group whose data we used, that we simply saw to few of these events and did not make any notice of when they happened. In their paper they did not specify how often these events occurred during their collecting period of two months, which makes it difficult to assess if we collected one or not. Another reason might be that these observations cannot be made at our scale and that the correlation simply does not exist for this magnitude of followers and retweets. Compared to our data that has users with an upwards boundary of about a hundred thousand followers Bakshy et al. has closer to four million.

As previously mentioned we can only collect a fraction of all tweets posted to Twitter. This, in combination with that there are significantly more users with just a few followers than there are with many, means that if selected at random, the chances for selecting one with many followers is very low. This might mean that Bakshy et al. were much stricter in regards to which accounts they collected data from. If that is the case they could have collected a very different kind of data set which also might explain why we do not see the same tendencies as them.

However, different studies have shown different results on this subject. Suh et al. [12] found that there is a correlation, but there are also other factors that have an impact, such as the age of a user’s account, which could make patterns less obvious. Cha et al. concluded that it was more the content of a tweet that affected its retweetability.

We can thus conclude that although follower count has an impact, there are too many other factors that weigh in to get a distinct, or linear, pattern.

How Follower Count Impacts Depth and Breadth
As earlier observed in Figure 4.14 when a user has few followers, their tweets reach deep compared to their breadth. This is plausible since the maximum amount of people that the
tweet will reach out to is fairly small. Therefore, if the tree is going to get large, it has to be in the form of depth. When looking at tweets where the original tweeter has many followers, the trend is not as linear. It is however necessary to pay attention to the vast decrease in the amount of data points here. We saw in Figure 4.13 that data points were more gathered at lower values for lower follower counts. This can be interpreted as larger follower counts meaning a wider spread, as could be expected. The Bitly links, contained within these tweets, reach a larger audience from the beginning. Therefore when someone posts a retweet it does not have the same impact as it does when the original tweeter only reaches a few people.

Retweetability Depending on News Source

We observed that there where differences in how much a link from a well known news source was retweeted depending on the source in question. This goes against the statement made by Suh et al. [12], as they claimed that the content of a URL included in a tweet did not matter as much as the fact that there was a URL included in the first place.

We saw that the highest average retweet rates were for articles from BBC, The Times and Fox News whereas the lowest were from CNN and The Guardian. Therefore, the statement made by Kwak et al. [13] that CNN and Huffington Post were among the most retweeted users, does not present itself in our study.

The fact that there are similarities between the variations in rates among groups of news sources shows that some of them probably share certain characteristics in regards to what articles they produce and how Twitter users respond to them. As an example, the high variations in retweet rate for BBC and The Times suggests that users retweet these sources very differently depending on for example the topic of the article or what user it was that made the tweet. The fact that their medians and their averages are alike shows that the news sources are similar in how popular they are to retweet from.

5.2 Method

Limitations of this study were predominantly due to limitations of the Twitter API, both in what data could be collected and the speed at which it could be collected at. Due to the Twitter API not supplying us with every single tweet being posted to the cite we are uncertain if we managed to collect all the data that was relevant to us. This is something that has to be taken into account. Alongside this the given constraints on how many requests that can be made per time period made all data retrieving take longer than desired. This limited our tree building process to sixty nodes per hour at the lowest, which means that doing this at a large scale is difficult due to limited time.

As the study proceeded, it became more and more clear as to why the data in a retweeted_status or quoted_status only referred back to the original tweeter. It would simply be impossible to, without storing massive amounts of data, tell exactly who exposed a user to a certain tweet as there are many ways of doing so considering the many functions of Twitter.

The majority of our time and resources were dedicated to handling the large amount of data obtained by the Twitter API. There are some factors that could have been handled differently in order to minimize this bottle neck and increase productivity. One of these factors is the way the Twitter streaming API works, something that we cannot directly control. This is unless an update would be released that made it possible not to fetch all of a tweet object when using the streaming API. Such an update would probably not happen as it only alleviates computing power on the client side but increases it on the server side. Another way of doing this is to narrow our downloading criteria. Since our only criteria for sorting out tweets was based on their content we could have applied more restrictions, such as original tweeter or time published. As we applied these restrictions at a later time regardless we could have ignored downloading the irrelevant data in the first place.
The way we save our data could have been done differently. The original code we built ours upon uses windows folder structure and txt-files in order to store data. Some form of database structure might allows for faster look-up times as opening and closing several file streams takes a lot of time.

In regards to creating propagation trees, we assess our algorithm to be precise to the extent that only factors from third parties are limiting its precision. If Twitter were to make more information about the relation between tweets, retweets and quotes available it would stand to reason that most of our heuristics for doing so ourselves would be made obsolete. This would allow a totally accurate representation as all sources of potential error would be eliminated. It has a high level of replicability as following the steps described in the method section should result in a comparable structure when creating other trees.

5.3 The Work In a Wider Context

When handling large amounts of personal data it is important to think about the ethical aspects of doing so. This is especially true when the data is obtained through social media as it can easily be linked back to the owner of the data. Personal information on the internet has been a hot topic for some time now and it is only becoming more relevant as the General Data Protection Regulation (GDPR) is about to come into effect in late May of 2018. The regulation intends to make it easier for the individual to control their personal data and how it is being handled [18]. With such a big change, however, comes a large amount of responsibility for all parties handling the data. As more trust is put into a system by the consumer not only is there an obligation to abide by the new rules, under the penalty of law, but there are also moral obligations to consider.

It is difficult to predict how GDPR will affect Twitter as it is new and the effects it will have on big companies such as Twitter is yet to be seen. If changes are made to the Twitter API to accommodate these new regulations it might have an affect on our work. Either by bringing on massive changes to the method or rendering it useless all together. Even if Twitter does not change their API we will still have to deal with the effects brought on by GDPR as we are handling prohibited data. Twitter might have their users agree to their data becoming public domain upon publishing it as a result of the regulation. If they do not however, we will have to ask each individual whose data we are using for consent, something that is not feasible.

It is important to note that our intent is not to be able to link data back to its owner. We are only interested in studying the propagation patterns in Bitly links and any other potential usage is a side effect of how the data is obtained from Twitter. As it is obtained in tweet objects that contain this information on the user, it is currently unavoidable to get personal data alongside the data we use for our study. If Twitter in response to GDPR decides to change how the data is requested, in order to distinguish personal data from non-personal data, we will probably with a few tweaks be able to employ our current method.
6 Conclusions

This study aimed to build a tool for creating tree structures that visualize how content is spread on Twitter. It also had the purpose of finding patterns in these propagation trees, for example in regards to how many followers a user has and in what way their tweets are shared by their followers. Another example of what could affect these patterns is what content is shared via the URLs posted and if that could have an impact on how the resulting tweet is shared. The tool was evaluated to have a high enough accuracy, and its limitations depended mostly on the request limits enforced by the Twitter API. The request limits made tree construction a time consuming task, which limited the scale of the study.

It could be concluded that most retweets are made within the tweet’s first twelve hours. New depths, meaning lengths of the chain from original tweeter to a retweeter, are reached fast compared to the rate in which retweets are made. When tweeting a link from a known news source, which source it is and what the article is about seems to have impact on how much it gets retweeted.

Most tweets do not propagate deeply, but widely. Those that result in a large tree usually have wider trees compared to their depth. For tweets where the original tweeter has many followers, the pattern is essentially the same. As expected, there is also a pattern suggesting that the more followers a user has, the more their tweets get retweeted.

Future Work

In order to make large scale observations about these propagation trees, a study should be made over a longer period of time to be able to process enough data. As this study mostly investigated the general attributes seen in spreading patterns it might be interesting to look into more details in the future. One could look into properties of the users that post tweets and see how their characteristics impact how content is spread. Another idea is to look deeper into what type of content is shared, and exactly what properties of a news article that make it interesting enough to retweet.
Bibliography


29


[18] Intersoft Consulting. “General Data Protection Regulation”. In: (May 2018). URL: https://gdpr-info.eu/