Inferring Interestingness in Online Social Networks

Will Webberley

2013

Cardiff University
School of Computer Science & Informatics

Declaration

This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed ..... (candidate)

Date .....

Statement 1

This thesis is being submitted in partial fulfillment of the requirements for the degree of PhD.

Signed ..... (candidate)

Date .....

Statement 2

This thesis is the result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by explicit references.

Signed ..... (candidate)

Date .....

Statement 3

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... (candidate)

Date .....

Acknowledgements

Abstract

Abstract text here etc.

Contents

List of Publications

Some of the work produced towards this thesis has also been published separately as follows.

- W. Webberley, S. M. Allen, R. M. Whitaker. *Inferring the Interesting Tweets in Your Network*, in *Workshop on Analyzing Social Media for the Benefit of Society (SOCIETY 2.0), 3rd International Conference on Social Computing and its Applications (SCA)*, Karlsruhe, Germany. *IEEE 2013*

- W. Webberley, S. Allen, R. Whitaker. *Retweeting: A Study of Message-Forwarding in Twitter*, in *Workshop on Mobile and Online Social Networks (MOSN'11), 5th International Conference on Network and System Security (NSS)*, Milan, Italy. *IEEE 2011*

List of Figures

List of Tables

List of Acronyms

OSN Online Social Network

MTW Mechanical Turk Worker

Glossary

Author

A user that has written a Tweet (*the author of Tweet* t *is denoted as* $A(t)$).

Follower

A type of user. A user, *x*, is a follower of user *y* if user *x* follows user *y*. Other users who follow a particular user will receive all of the user's Tweets and retweets to their home timeline. A user can elect to follow another user.

Friend

The inverse of follower. User *x* is a friend to user *y* if *y* follows *x*.

Path-length

The penetration of a Tweet - i.e. the number of times a Tweet is retweeted down one chain. The final retweeter in the chain indicates the number of hops the Tweet has taken from its author.

Retweet

n. - A replica of a Tweet, which has been forwarded on by a user (who is not the Tweet's original author) to their own followers (*denoted as* $RT(t)$).

v. - The act of replicating a Tweet. A user who finds a Tweet interesting may retweet it so that it gets more exposure.

Retweet Group

Set of Twitter users responsible for the propagation of a Tweet. Comprises the original

author of the Tweet and the users which have since retweeted it.

Retweet Count

The number of times a particular Tweet has been retweeted.

Timeline

A collection of Tweets in Twitter in reverse-chronological order. A **user** timeline consists of that user's Tweets. A user's **home** timeline consists of the Tweets of each friend of the user.

Tweet

n. - A piece of information in Twitter; a piece of text, less than 140 characters long, which is written by a user. When sent, the Tweet is pre-pended to its author's user timeline and also to the home timelines of each of the followers of the Tweet's author (*denoted as* t).

v. - The act of writing and sending a Tweet.

Note - A Tweet, in the context of Twitter, is treated as a proper noun and as such has its first letter capitalised[1].

User

An account on Twitter. Each user (usually representing a real-life person or organisation) can Tweet, retweet, follow other users and be followed by other users. In this thesis, sometimes the terms *user* and *person* are used interchangeably.

[1] https://twitter.com/logo

Chapter 1

Introduction

Online social networks have exploded into the lives of millions of people worldwide over the last decade, and their use has dominated the communication highways and facilitated the interconnection of the world in ways never before perceived possible.

These social networks imitate real-world social networks. Although most such platforms each provide a different service to collaboratively satisfy an array of different use-cases, they tend to all be based around the idea of 'friendships' (i.e. links between the user nodes in the social graph) and the sharing of information amongst friends.

Social networks like these have been available for around ten years now (with MySpace[1] launching in 2003 and Bebo[2] in 2005), but it wasn't really until Facebook's[3] worldwide launch in 2006 that social networks became the staple, ubiquitous norm that they are today. More recently, we have seen the introductions of Google's social network grown from its Buzz service, Google Plus[4], Pinterest[5], App.net[6], and many more. They make up a large part of the basis and meaning behind the ideas of Web 2.0, which describes the web as being primarily formed from user-generated content and encourages the sharing of such content.

Another component that helped in the dawn of Web 2.0 was the rise of *blogging*. A

[1] http://myspace.com
[2] http://bebo.com
[3] http://facebook.com
[4] http://plus.google.com
[5] http://pinterest.com
[6] http://app.net

blog ('web-log') is a time-based series of posts consisting of continuous pieces of text, photos, or other media, and is generally contributed to by a single author. Blogs are often based around one or a set of topics and are usually public - meaning that they are written with the intention of being read by others. Despite this, they are often a way in which the author can look back at their history of posts, acting more as a diary recording snapshots of the author's life.
Various blogging services exist on the web today, such as Medium[7], Wordpress[8], and Tumblr[9].

1.1 Twitter as a Social Network

Twitter[10] is an online social network, which launched in the summer of 2006 [**?**]. Since then, it has rapidly gained in popularity amongst several different user groups - teens and young people, casual users, celebrities, reporters, and so on - and within eight months had around 94,000 registered users [**?**]. Although Twitter has never been a direct competitor with Facebook, users tend to use the two sites concurrently for different purposes: whilst Facebook's focus is on providing many services at once (such as photo-sharing, commenting/endorsing of information, messaging, pages for businesses, groups, events, etc.), Twitter's is more on simplicity.

More specifically than just being an online social network, Twitter is a microblogging website. Whilst a blog, as mentioned, typically contains long posts, Twitter only allows its users to post short pieces of text, up to 140 characters in length [**?**] [**?**], called 'Tweets'. Thus, Twitter is a hybrid social network and blogging service and whilst each Tweet may only realistically be able to hold a couple of sentences, this system facilitates quick, timely, and 'real-time' *live* information-sharing amongst its millions

[7] http://medium.com
[8] http://wordpress.com
[9] http://tumblr.com
[10] http://twitter.com

of users [**?**]. Its idea is that short pieces of news will 'travel' faster and will be seen by more people more quickly than traditional news stories.

Although Tweets are limited to 140 characters in length, the inclusion of URLs is allowed. This enables further extension of Tweets through external websites, and supports the inclusion of links to images and videos. Twitter has encouraged this use-case by providing 'share' buttons for developers to embed in websites, and direct support for photo and video applications, such as TwitPic[11] and Vine[12].

Its simplicity has also helped its growth into the mobile domain, in which smartphone users are able to very quickly post updates about their lives, a piece of information they want to share, or a photo or video, and be able to post it *as it happens* directly from the news source or geographical location [**?**]. This has been especially useful in emergency situations worldwide, including the Haiti earthquake in 2010 [**?**], and 2011's Egyptian protests [**?**] and Thai flood [**?**].
Indeed, [**?**] used Twitter to build an earthquake-reporting system for Japan that outperforms the Japan Meteorological Agency in terms of its promptness of notification.

Use of Twitter is based around 'timelines' of Tweets, to which new Tweets are prepended as they are posted by users. The *home* timeline is the default view, in which Tweets from all of a person's subscribed-to users are placed. Timelines of an individual user contain only Tweets from that user, and are known as a 'user' timeline. Customisation of timelines is also possible through the use of Twitter lists, in which different users can be placed to categorise streams of Tweets from different sets of users.

[11] http://twitpic.com
[12] http://vine.com

1.2 Twitter's Social Graph and Information Subscription

As with many social networks, the structure of Twitter lies within the users and their connectivity within its social graph. However, unlike Facebook, whose social structure is made up of bi-directional 'friendships' between users, Twitter's primary social graph is made up more of mono-directional links between its users [**?**]. A person using Twitter can elect to *follow* another user, which subscribes the person to receive all of that user's Tweets to their home timeline. The set of users that follow a person are known as that person's *followers*, and the set of users that the person follows are the person's *friends*.
Therefore, if two users both mutually follow each other, then the link between them is bi-directional.

Whilst bi-directional links are common amongst communities of similar interests, friends, colleagues, and so on, mono-directional links are found more in situations in which less-influential users follow more-influential users, such as celebrities.

1.3 The Problem

A user who follows a set of other users can *generally* be said to find that set of users to produce more interesting information than those users that the user does not follow. However, despite that, not *all* information produced by an 'interesting' user is likely to be interesting, and yet *all* information produced by a Twitter friend will be received onto the home timeline.
Noise is a common problem in Twitter, and is the uninteresting information one might receive that conveys little interest. It is likely that most of the information received on Twitter *is* uninteresting [**?**], and this makes it very hard to distinguish the interesting information from the uninteresting.

Since people tend to use Twitter most in short sporadic moments, looking for a quick news fix, they do not have time to filter out noisy information. Thus, the presence of noise can dampen the experience of the user, making it much more difficult to find interesting information.

In addition, Twitter users typically exist within an information 'bubble'. This is similar to the notion of the Google search bubble, in which the search engine uses previous results and search terms to only return information to a user based on what *it thinks* the user would find the most interesting and useful.
This results in the users not knowing which information exists beyond the confines of their bubble, and if they do not know it exists, they cannot know if it is of interest to them. Similarly, a Twitter user cannot follow all of the users he/she may find interesting, since he/she will not *know* of all the interesting users existing on the social graph.

How can users be exposed to *interesting* and *relevant* information, but without them having to know about it or look for it first?

1.4 Contributions

This thesis focuses on understanding information propagation, and how this combined with knowledge of the social structure of Twitter can assist towards solving the problem of identifying interesting and relevant information and determining it from the noise on Twitter. Whilst other work in the area has also looked into the notions of relevance and interest in online social networks, and Twitter in particular, none has addressed the problem in such a way as this.

Part of the outcome of this research are methods for effectively inferring interesting information and, indeed, ranking information by interestingness. The methods are validated in various ways to help highlight their strengths and weaknesses in performing inferences and appropriate use-cases.

The work addresses the problem area in that it helps towards solving the goal of identifying *globally* interesting information in Twitter. In addition, certain measures are taken in an attempt to address the idea of information relevance, which denotes how information interestingness is subjective, and thus different from user to user.

1.5 Thesis Structure

The rest of this thesis is structured as follows.

A background is provided as an introduction to some of the ideas behind the main research, which immediately follows this chapter, and includes a review of relevant literature across the range of topics addressed in the thesis.
Following this are chapters that contain research on Twitter's information propagation characteristics and its interesting and useful behaviours, the social structure of Twitter and the ways in which this is important for understanding the spread of information, and then on the research of the methodologies themselves, including validation and analysis of the results of this work.

The thesis ends with a general analysis and conclusion, and a discussion of potential future work in this area and leading on from this research.

Chapter 2

Background

One of the most widely-used features of Twitter is its inbuilt function for facilitating the spread of information within its social structure. This phenomenon is the basis for much of the research in this thesis and, when combined with the characteristics of Twitter's user graph, has many interesting attributes and behaviours associated with it.

2.1 Domain Context

2.1.1 Information Propagation through Retweeting

The function of propagation in Twitter is known as *retweeting*, and is carried out by the Twitter users themselves. When a user views a Tweet that they believe to be particularly interesting, and believe it to also be interesting to his/her followers, then he/she can elect to retweet it, and thus pass it further through the social graph to that user's followers also. A Tweet that has been retweeted is known as a *retweet*, and it is clear that a Tweet which is retweeted will be made available to significantly more users than a Tweet that isn't retweeted [**?**] [**?**].

Since Twitter's social graph is decentralised and retweeting occurs between individual groups of users, its properties are similar to information dissemination in other types of decentralised graphs, such as content-forwarding in opportunistic networking [**?**].

A retweet can be carried out in one of two ways: either through the use of Twitter's native retweet button, or manually.

The retweet button is displayed along with each Tweet in a Tweet timeline which, when clicked, immediately creates a new retweet containing the verbatim content of the original Tweet and automatically sends it on to the retweeter's followers.

The user who created the original Tweet is credited as the author on the recipients' timelines, with an indication of who carried out the retweet itself. Thus, users who follow the retweeter will see a Tweet appear in their home timeline from someone that they may not directly follow.

Figure 2.1: A retweeted Tweet

The manual approach involves physically copying the content of the Tweet to be retweeted and pasting it into a new Tweet, usually with the text '`RT @<username>:`' pre-pended, where `RT` stands for **re**tweet and `<username>` is the username of the author of the original Tweet. This method allows for annotating the original content of the Tweet (for example, to provide an opinion on the Tweet contents), producing a *modified* Tweet, which can sometimes be pre-pended with `MT` rather than `RT`.

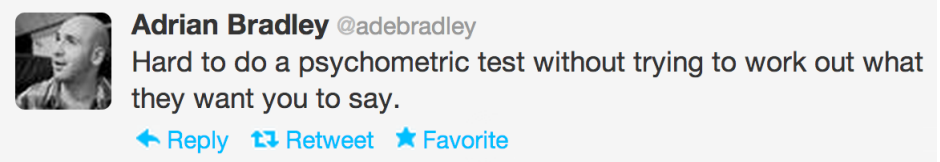


Figure 2.2: The retweet 'button' in context

Each Tweet has a retweet count associated with it, which is the raw representation of the number of times that the Tweet has been retweeted using the retweet button method.

Since the manual retweet technique is more community-driven, there is no official way to include these as part of the retweet count of the original Tweet. However, since the manual method is typically only really used with the aim to annotate or modify the Tweet in some way, the resultant 'retweet' is no longer a real representation of the content of the original Tweet anyway, and so should not be counted as such.

It should be noted that Twitter users may choose to make their account 'protected'. A person who has a protected account will still have a publicly-visible profile (displaying a name, username, bio, and so on), but their Tweets and other information (such as the followers and friends lists) are hidden from users that aren't followers of the person. Potential followers of a protected account must *request* a followship, which can then be accepted or rejected by the protected account holder.
Since Tweets from a protected account are only visible to approved followers, the retweet button is unavailable for them to disseminate the Tweet any further than the author's immediate local follower network. However, since the manual retweet method does not rely on the button and isn't governed by Twitter, a protected account's Tweets can still be retweeted in this way.

In a similar way to Facebook supporting the endorsement of information found on its site by inviting users to 'like' a piece of content, retweeting is effectively a *vote* or endorsement for a Tweet on Twitter. In both cases, the number of likes and number of retweets is visible to the platforms' respective users, and so this provides some insight into the *popularity* of the information.

2.1.2 Retweets and the Social Graph

The social graph of Twitter is constructed, like in other online social networks, by edges between users, partially emulating real-life social interactions between humans. The growth of social media has encouraged more dense communication between users

all over the world, who would not previously be able to be in direct contact with one another in this way.

Derived from this, Stanley Milgram's finding of "six degrees of separation" [**?**], which defines that people are usually no more than six hops away from each other on the 'real-life' social graph, was found to be an overestimate when it comes to the analysis of the structure of OSNs by [**?**], who found that the average 'distance' observed in Facebook's entire 721 million-node graph in 2011 was only around 4.7 hops. This implies that denser links between users and larger communities that apparently manifest themselves in OSNs create a smaller 'world' than that experienced in reality.

In each of Milgram's experiments participants passed a message to one another, at each stage only passing to other people that they actually *know*, in the hope of the it reaching a single intended recipient. This meant that people could use acquaintances in other geographic locations to transfer the message from community to community.
Twitter supports a similar propagation mechanism in the fact that retweets can themselves be retweeted; this is a focus of some of the earlier research in this thesis.

This behaviour provides further penetrative 'depth' of the information through the social network away from the source user in addition to the spread 'width' made by the initial retweets. Although retweeting is not carried out with the aim of information reaching any particular final user (or set of), as with Milgram's experiment, this phenomenon allows retweets to 'travel' between 'online communities' of users.

As with real-life social networks, communities of users in OSNs are also a common feature [**?**].
In Twitter, these communities are typically small to begin with and are based on a topic of interest or around a more influential user. As more Tweets are produced from within the community, further links are made to interconnect the community's users, producing a growing 'swarm' of interest around the initial topic or user [**?**].
As further users begin associating themselves with this community, its audience becomes more widespread and the community grows. This concept is discussed in greater

length by [**?**], who also experiment further with communities and describe them as compact groups of users connected by dense follower links.

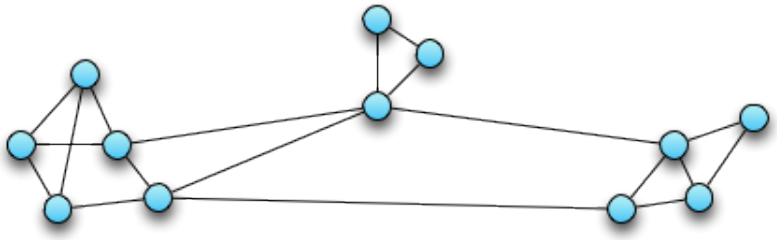


Figure 2.3: A hypothetical group of user communities

In more dense communities, Tweets can be made available to many users immediately after they are published, since many of the links between users are shared. This means that any retweets that occur within communities are likely to have a lot of *redundancy*, in that many of the retweets will be sent to users who have already seen the Tweet. Although Twitter prevents this information duplication by not showing the retweets of Tweets that have already appeared on a user's timeline, it does increase the chance of the Tweet making its way out of the community.

Retweets amongst users within a community are likely to be common, due to the shared-interest nature of communities, and some users can provide 'bridges' by being active in more than one community. In these cases, Tweets can be passed between the communities through retweets by the bridging user. If there are many users sharing communities, then there are many more avenues available for propagation to occur down, causing a high level of information throughput. If there are fewer bridges, then there is more of a bottleneck between the communities, hindering the information spread.

[**?**] also finds that communities can be formed from different types of people, such as those who Tweet frequently and have many followers, and those who contribute very little and have few followers. Those with many followers and many friends receive lots of information and have the potential to spread information further than those with

fewer inward and outward edges. Studies in the behaviour of different types of users in Twitter is done more thoroughly in [**?**], which defines 'broadcasters' (users with many followers and few friends) and 'miscreants' (users with few followers but many friends) and their roles in information propagation.

Users that retweet the interesting information from a source user to others, who do not follow the source user and so would not naturally receive the information, are effectively acting as information *filters*. By not following the source user, a person might still receive the interesting information through these filters, but will not receive any of the 'noise'. Thus retweeting means that friends of a user become useful filters of information for users further 'downstream' and retweeted information can be said to have a higher *credibility* than Tweets that aren't retweeted [**?**].

2.1.3 User Influence

Just as there are different types of user *behaviours* on Twitter, as mentioned in the previous section, the are also users of different *influence* levels [**?**].

Much research has gone into user influence, including on how this might be detected [**?**], and influential users are generally found to be those that have a greater impact on Twitter's social network [**?**] and that usually have significantly more followers than an average user. Influential users tend to have a high persuasion over other users, relating *influtentials* in Twitter to those who are also influential in the real world as part of traditional communication theory [**?**], and therefore many Twitter influentials are the accounts belonging to real-world celebrities.

As with real-world celebrities, Twitter influentials are those with many 'influenced' followers, or fans, which are the users who have the strongest agreeable opinions of the influential. As a result, an influential user has a greater number of followers who are interested in the information produced by the user, and is therefore more likely to receive more retweets than less influential users.

Although influence level is partly derived from the follower count of the user, it should be noted that a user with high in-degree on the social graph[1] does not necessarily imply a high level of influence. An 'active' audience of users who reply, retweet, and interact are more indicative of an influential user [**?**]. This is especially true since a user can gain more followers through campaigns such as '#teamfollowback'[2] or by following 'out of politeness', in which a user will follow another user back as an act of politeness, but these users tend to have *both* high in- and out-degree and invoke less interactivity amongst their followers, which are not necessarily characteristics of an influential user [**?**].

Klout[3] is a web service that attempts to review a user's social media influence by assigning users a Klout Score. Their website declares that this score, which ranges from 0 to a maximum of 100 and whose generation algorithm is kept private and unpublished [**?**], is determined from a variety of 400 sources taken from eight different social media platforms, and which *also* seems to take interactivity between users as the primary indicator [**?**]. Additionally, the service indicates the topics a user is influential about, with the general idea being for organisations to check up on which users are influential for marketing purposes, but also to highlight the users that should be replied-to at a higher priority.

2.1.4 Twitter as an Information Retrieval System

From a high level, Twitter is essentially just a variety of information-retrieval system, which people can utilise to produce and consume information when required. In traditional information-retrieval systems, such as search engines and library systems, keywords and search terms are common ways for describing the type of information the user would like to receive back. The system would then search a database or archive

[1] In-degree: many followers

[2] Users associate themselves with #teamfollowback to imply they will return all followships.

[3] http://klout.com

for what it believes is relevant information, *based* on these 'retrieval parameters', and return results to the user ordered usually by the estimated relevance of the articles [**?**].

Information quality is also reliant on the expected reading effort of the returned documents. The character precision-recall metric was introduced by [**?**] by way of demonstrating the tolerance-to-irrelevance ratio. The general mechanism for this ratio is to do with users reading a document passage; the point at which this ratio is reached is when the user stops reading the particular passage and moves to the next whole document, since they assume the rest of the document is also irrelevant to them.

Therefore, the more effective the information retrieval system is in displaying high-quality information, the lower the chance that this ratio is reached by the user.
It is comparable that a Twitter user viewing Tweets from a user they are following may get to the point where he or she reaches this ratio (i.e. is beginning to get bored or find the Tweets irrelevant) and decides to unfollow the friend. Similarly, the more effective the user is when selecting people to follow in the hope of receiving interesting information, the less likely it is that the user will remove these friends.

Whilst Twitter does not support the use of keyword searching for its primary information delivery method, it does lend its users some control over the type of information they wish to receive. As mentioned previously, users receive all of the Tweets from everyone that they follow onto their home timelines. Thus, by selecting users to follow, a person is effectively describing and implicitly indicating the type of information he/she would like to receive, and by editing their friends list (either by adding new followers or pruning existing ones) he/she can alter this indication.

Despite this control, it is still unlikely that users will achieve a perfect Twitter experience due to the presence of *noise* [**?**]. As discussed in the Introduction, this problem stems from that although a person follows users they consider to be interesting, it is often the case that not *all* information produced by interesting users will be interesting itself.

2.1.5 Information Quality, Popularity and 'Interestingness'

Information-retrieval systems typically use some measure of information *quality* when determining which documents to return to a user and also when deciding on the *order* the documents should be displayed in. This 'quality' is subjective in that different systems use a variety of different algorithms for deducing quality, usually based on the level of *interest* in each of the available documents (such as Google's Page Rank algorithm and Amazon's recommendation algorithms), but also in that the level of quality itself depends on the user itself requesting the information.

In the case of Google's Page-Rank, the algorithm uses multiple cues to determine who the user is, their interests, past searching habits, links clicked, and so on, to return *relevant* information, which is incidentally one of the causes of the aforementioned Google search bubble.
Amazon's recommendation algorithms analyse a user's past item views and purchases and cross-matches these against trends based from users who also looked or bought similar items. Amazon is then able to accurately determine the type of items a customer are interested in purchasing, and can send emails to that customer with personalised recommendations.

Thus, information quality is essentially a function of information interestingness and information relevance, which are both related to the concept of *effective stimulation* [**?**] discussed later.

Twitter uses no such metrics to deliver information to its users, relying on the users themselves to implicitly 'choose' the information they want to receive - it is an information retrieval system and not a recommendation system. Additionally, information is always displayed in chronologically-ordered timelines, with new Tweets being continuously inserted at the top as they occur. Twitter does not try to indicate interesting Tweets on the timeline which means that the interesting information is shown at equal value alongside the 'noisy' Tweets, causing the difficulties in identifying the interest-

ing information as has been mentioned previously.
Indeed, the recent TechCrunch article from October 2013, "Twitter Quitters And The Unfiltered Feed Problem"[4] talks at more length about this particular phenomenon, and helps highlight the problem area of this work more clearly.

The retweet count of a given Tweet is a useful metric in inferring a Tweet's *popularity*. If a Tweet is retweeted 10 times, then ten people have taken the time to read that Tweet, decide it is worth sharing, and then actually retweet it [**?**]. This user (and the other nine retweeters) may have found the Tweet interesting, yet it should be noted that although the count can be used as a measure of popularity, as a function of the influence of the Tweet's author, the retweet count alone cannot be used as a measure of how interesting the Tweet actually is [**?**]. For example, it is inappropriate to say that the first Tweet in Figure 2.4 is so significantly more *interesting* than the second, although it is clearly more popular since Justin Bieber is an extremely influential Twitter user.

Figure 2.4: Example of Tweets with significantly different retweet counts

Whilst the work in this thesis does not aim to build an accurate retweet-predictor, this does become a basis for some of the work in later chapters.
[**?**] identifies the same problem of 'noisy' Twitter timelines and discusses methods for predicting *popular* Tweets using a J48 decision tree classifier, based on the likelihood

[4] http://techcrunch.com/2013/10/05/sorry-my-feed-is-full

of the Tweet being retweeted by a particular user. Although the authors address information relevance from a user-centric point of view, the validations of whether a prediction of a retweet occurring for a given Tweet is actually indicative of the *interestingness* of said Tweet do not perform particularly well.
A retweet-prediction model based on a factor graph model is introduced by [**?**] to determine how retweetable a Tweet is on a global scale. A precision of just under 29% is achieved in predicting if a Tweet will be retweeted, but no mention is made of how this relates to how *interesting* the information is.
Another study into retweet prediction was carried out by [**?**], in which a trained probabilistic collaborative filter model (named 'Matchbox') was used to determine the useful features in making the predictions. As with the previous study, the research focuses on a retweet *probability*, which is a binary decision made by one particular user. The methodology is not aimed at the inference of interestingness, and simply determines that the most relevant features for accurate decision predictions are the author of the original Tweet and the retweeter.

Inversely, [**?**] and [**?**] predict the *type* of messages that are likely to be retweeted further, the latter using a logistic regression to both predict an individual retweet decision and a retweet *volume*. The methods do not apply these notions to how interesting the information actually is, achieve low recall and the multi-classifications seems only to perform well on very unpopular or very popular Tweets. It is made clear, however, that the retweet volume of a Tweet is useful in denoting Tweet *popularity*.

[**?**] uses a passive-aggressive machine-learning algorithm to make binary predictions on retweet decisions and cited that social features - for example, number of followers of the author, frequency of Tweeting, etc. - were the largest factors in the performance, and [**?**] uses a logistic regression, partly using a dataset published as part of another paper by the same authors as [**?**], to predict retweet decisions in order to address information interestingness. However, little effort is made to define interestingness or, indeed, validate that the inferences towards this are accurate and correct.

A logistic regression is again used by [**?**] for predicting binary retweet behaviours with the focus on information propagation in disaster scenarios, and [**?**] showed that conditional random fields can perform better than logistic regressions than when modelling retweet behaviour in the same way.

Since the above papers only effectively consider a prediction of retweet outcome, which is a binary decision, it is hard to relate this to more of a global interestingness, aside from stating that a retweet implies the retweeter's relative interest in the Tweet. However, a retweet count, as mentioned above, is inappropriate as an indicator of *magnitude* of interest, and so the research into predicting individual retweet decisions cannot be used as a basis for this. Additionally, not much emphasis is placed on how well the techniques work 'on-the-fly'; many of the methodologies discussed require several features that may take a long time to collect and compute, making them unsuitable for use as part of quick and useful interestingness evaluations.

The idea of Tweet scoring and retweet *count* predictions is introduced by [**?**], who used their methodologies to produce a system[5] enabling users to compile Tweets in ways that are predicted to achieve the most retweets. The predictions are based on averaging the score, derived through a linear regression, of different components of a user's Tweets (such as the inclusion of a particular hashtag), so that when a Tweet by the same author is next constructed, the various components of the new Tweet can be compared against the scores of the counterparts seen in previous Tweets. The value produced through this method is then used to generate an expected retweet count as part of a comparison to the user's average ('baseline') achieved retweet count at this point in time, and was shown to perform well on influential Twitter users.

However, the methods described do not take into account fluctuations in the social graph, particularly in the case of less-influential Twitter users, who's local networks are prone to more frequent changes. Additionally, they rely on enough previous Tweet and temporal information on the user to be evaluated, and do not relate the resultant score

[5]https://sites.google.com/site/learningtweetvalue/home

to any type of interestingness metric in the context of highlighting it from amongst noise.

Alonso et al. ([**?**]) also use 'scoring' to address interestingness, focusing more on determining *uninteresting* content, by assigning Tweets an integer score out of five. Although the authors initially attempted to train a decision tree classifier on a set of 14 features, they settled on classifying a Tweet as 'possibly interesting' if it simply contains a URL, and otherwise classify it as 'not interesting'. Although the authors did then further classify the possibly interesting Tweets, by studying the magnitude of the crowdsourcees used to evaluate the Tweets that found them interesting, and then classifying Tweets based on them containing a particular type of named entity - for example, a person's name, a place or brand name, and so on - the categorisation system is too coarse and is not capable of representing the many different types of Tweets seen on Twitter.
Additionally, despite achieving relatively high accuracy in this particular area, the methods are not suitable for assessing Tweets on a general or user-specific level, especially since Tweets that don't contain URLs might still contain interesting content.

An interesting study is described by [**?**], in which a clustering algorithm is used, taking into account the retweet count of a Tweet and how this is related to the popularity of the source user, to determine information quality. Although this work is more similar to the research discussed later in this thesis than others, the scoring is quite simple and the author's use-case seems limited to that of identifying the most important Tweets surrounding a particular event (such as the death of Michael Jackson).
Additionally, the authors do not make any effort to verify their results in any way, aside from comparing the Tweets determined to have a high quality by each of their two assessed methodologies.

2.1.6 Precision and Recall

Precision and recall are two metrics that are often used simultaneously to verify the performance of a method or procedure, with the usual goal being to maximise both. The metrics are used for validating *accuracy* in different ways, yet they can be applied to other purposes also and are useful in describing the notion of interestingness in Twitter.

The precision and recall measures are talked about somewhat in Twitter- and retweet-based literature. These pieces tend to only analyse the measures on their own work when applied to Twitter rather than on any more global scale. Certainly, there is less in the literature on the subjects of precision and recall with regards to retweeting in general.

The idea of assessing the credibility of information is introduced in [**?**], in which the authors demonstrate methods of measuring the credibility of 'news' and 'chat' Tweets. In this case, retweeting is seen as a possible measure of a Tweet's credibility, since users typically only retweet information they see as interesting or useful. The authors use a logistic regression on a set of features derived from each Tweet in order to classify its credibility.

The precision and recall metrics are used to verify the different aspects of the paper's results. In particular, they are applied to the classification of assessing credible information (and users) in order to calculate how well classified the information is. A higher precision, therefore, shows that their model has accurately classified most of the total information classified as either credible or non-credible.

$$Precision = \frac{\text{Number of correct classifications}}{\text{Number of total classifications made}}$$

$$Recall = \frac{\text{Number of correct classifications}}{\text{Total number of potential classifications}}$$

On a similar note, [**?**] discusses the notions of precision and recall more generally. The authors discuss the problem regarding the balance of information received by Twitter users. Having too few friends reduces the number or interesting posts received (i.e. low recall); having too many friends may cause information overload and is likely to include a lot of noise (i.e. low precision). This issue is used, instead of to validate results, as a basis for the work; predicting the Tweets that are most popular and will be retweeted the most.

In addition, precision and recall are used to compare the method to two other baselines; the TF-IDF score and *Retweet Before*, which uses the fact that if a Tweet in the training data has been previously retweeted, then it's likely to be retweeted again. The two metrics are also used to compare results when certain features are removed from the classifier. For example, showing that without using a 'user retweet' feature, the precision and recall remain significantly higher than when removing other features, meaning that this feature does not contribute highly to the performance.
More specifically, precision and recall are used in a similar way to in [**?**]; except rather than looking at the number of classifications made, the authors use the number of predicted retweets.

[**?**] discusses a proof of concept for detecting influential users in one of two categories; evangelists or detractors. Precision and recall, in this case, are used slightly differently:

$$Precision = \frac{\text{Number of influential users retrieved}}{\text{Number of users retrieved}}$$

$$Recall = \frac{\text{Number of influential users retrieved}}{\text{Total number of users}}$$

The concept is taken further through the use of another metric, the *Mean Average Precision*, which is used to denote an influential user as being a detractor or an evangelist. A high precision, in this case, would imply a large proportion of influential users are classified correctly and a high recall means that most of the influential users existing in the entire dataset have been classified. The final results then show the precision and

recall values for detecting evangelists and detractors in both follower/following networks and interaction networks. Both precision and recall improved when the size of the set of highest classified influentials increased (i.e. the top set of influential users).

[**?**] presents a method for the automatic classification of Twitter information to determine if a document is positive, negative or neutral in sentiment. In this case, the authors replace precision with *accuracy* and recall with *decision*, since they are using many classes instead of a binary classification, and define them as the following:

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Number of all classifications}}$$

$$Decision = \frac{\text{Number of retrieved documents}}{\text{Number of all documents}}$$

The accuracy is measured across the classifier's decision, and the $F_{0.5} - measure$ is then calculated based on these values instead in order to show that the classifier works well when the dataset size is increased.

As well as a good news source, Twitter is also used as an informational, user-contributed source on world events. [**?**] introduces a system, TwitInfo, which can be used for detecting, summarising and visualising events from Tweets. The authors looked at football match footage, web content, and earthquake survey data, and manually annotated major events in each to produce ground truth sets. These would be use to compare and contrast the results produced by their event detector using the following definitions of precision and recall:

$$Precision = \frac{\text{Number of events detected were from ground truth set}}{\text{Total number of events}}$$

$$Recall = \frac{\text{Number of events detected}}{\text{Number of events in ground truth set}}$$

With these definitions set, the authors were then able to easily calculate precision and recall for their algorithm.

For the work in this thesis, interestingness of information is the performance metric used to describe information quality, and thus precision and recall for any particular user in the scope of this thesis can be defined as follows:

$$Precision = \frac{\text{Number of interesting Tweets received}}{\text{Total number of Tweets received}}$$

$$Recall = \frac{\text{Number of interesting Tweets received}}{\text{Total number of all interesting Tweets}}$$

where *received* means that the Tweet has arrived on the user's home timeline, but does not imply that the user has *read* the Tweet.

Therefore, a user following many other users will receive lots of interesting information onto their home timeline in amongst lots of noise; resulting in a reduced precision and higher recall. Another user might follow a very select few other users who are of direct interest, and thus will experience high precision, but low recall.
These metrics are therefore useful in describing the concepts of noise and interestingness, and are consistent with their respective definitions in that users will achieve an optimum Twitter experience if both precision and recall are maximised.

Zadeh et al. ([**?**]) defined bespoke definitions of precision and recall, yet also in the domain of interesting information on Twitter. Although the authors identify the need for users to be able to discover other users of interest and declare that Twitter does, in fact, have a 'high precision' of interesting information, they admit to using a very coarse set of possible interest categories and is only based on *overlapping* interests rather than addressing the interest-noise ratio more concerning the research in this thesis. Additionally, clicks on URLs by users are the only means by which to measure this interestingness, and Tweets with URLs are usually the most interesting type of information [**?**].

2.2 Collecting Twitter Data

Most of the analytical work in this thesis relies on various data being collected from Twitter. Twitter provides an API for developers in order to facilitate the production of applications for its platform, but also for research purposes. It permits interfacing with many components of Twitter's service, such as posting and retrieving Tweets, interacting with other users (e.g. creating new friendships), and most of the features that Twitter's service itself provides to its users.
The API encourages use of the OAuth[6] authorisation framework to handle access[7], allowing Twitter to keep track of applications and each application's access privileges and rate limits[8].

Twitter's traditional REST API, v1[9], provided many useful endpoints for data collection and allowed each OAuth-authenticated application 350 hourly POST and GET requests[10].
In June 2013 Twitter officially deprecated v1 of its REST API, forcing use of its new v1.1 API[11]. The new version contains many of the same resources[12] as the original, but workarounds are required to get the results as some of the endpoint requests possible through v1. Additionally, new rate-limit policies were introduced, allowing more limited and controlled access to most of the available resources.

Since the work in this thesis was ongoing over this switch-over date, the initial work utilised API v1, and the latter work API v1.1, causing some changes to some of the data-collection methodologies as the thesis progresses. Descriptions of the data-collection in each relevant part of the thesis reflect this change, where appropriate.

[6] http://oauth.net
[7] https://dev.twitter.com/docs/auth
[8] https://dev.twitter.com/docs/rate-limiting/1.1
[9] https://dev.twitter.com/docs/api/1
[10] https://dev.twitter.com/docs/rate-limiting/1
[11] https://dev.twitter.com/blog/api-v1-retirement-date-extended-to-june-11
[12] https://dev.twitter.com/docs/api/1.1

2.3 Research Motivation

The motivation for the work in this thesis lies in the need to distinguish interesting information from noisy Tweets in Twitter, the latter of which is the problem area identified over the previous sections of this thesis.
It has been made clear that the retweet count of a Tweet cannot reliably be used as a measure of interestingness, especially in the context of influential users, who naturally achieve significantly more retweets than average users, but which does not imply that the information they produce is of a higher quality or interest level.

As a result, the retweet count alone cannot be useful in distinguishing interesting information from noise in a timeline of mixed Tweets from different users with different levels of influence - some further metric is required to make this distinction.

This thesis covers the procedure and research behind a methodology that determines and ranks information on Twitter through inferences of interestingness that allows the more interesting information to be brought forward.

Chapter 3

Understanding The Behaviour of Retweeting in Twitter

It has been discussed that the popularity of information in Twitter can be related to the propagation characteristics of that information through Twitter's social structure. That is to say, that the more times a Tweet is retweeted by users, the more people have found the information contained within it to be interesting enough to be worth sharing.
It has also been shown that this retweet count metric alone cannot be an implication of the actual interestingness level of a Tweet. This is related to the notion of user influence, which dictates that some Tweets are naturally immediately seen by more people in the first place and thus have a higher chance of achieving a retweet as they are. Indeed, the authors of [**?**] demonstrated that a user's Tweets' retweet rates increase as the user's follower count increases.

The strength of Twitter lies is in its social structure, where users can elect to follow and unfollow others as they desire and with immediate effect. Followers of a user receive all of that user's posts onto their individual (or 'home') timelines. If a user has set their profile to be public, then their posts also historically appeared on the *public* timeline, which is now deprecated, but was accessible to anyone; even those without a Twitter account. As a result, people are likely to follow users who update with interesting posts; whether the follower is a big fan of the user and simply wants to know everything going on in their life, or if the follower is simply interested in the topical area of most of the friend's posts.

Just as Twitter users will post Tweet with subjects that are of interest them - possibly related to a user's work, a hobby, or a mixture of multiple areas - and these Tweets are generally posted with the idea that they will be useful or interesting for some of their followers as well as an attempt to attract more followers, retweets are generated with the same motives in mind. This means that if a Tweet is retweeted, it is not only allowed to disseminate further through the social structure, but also that a higher Tweet quality is implied.

Thus, this describes how a user's friends, who carry out most of the retweets of the user, effectively become filters of interesting information for that user and for the followers of those friends. The *audience* of the original Tweet is therefore significantly increased. Since retweets are usually always attributed to the original author then you, a Twitter user, may gain more attention by means of followers by posting *interesting* Tweets, which will;

1. increase the chances that users reading your Tweets will choose to follow you, and;

2. increase the chances that users will decide to retweet your Tweet, thus broadcasting it to a larger audience. People viewing this *retweet* then may decide to follow you.

Since a Tweet can be retweeted multiple times, and, as mentioned, a retweet itself can also be retweeted, the much larger the effective audience (both directly and through retweets) of a Tweet's original author has the potential to become if they choose to post interesting information. In this chapter, an understanding of the behaviours and properties of retweets is provided, along with discussions into how these are relevant in researching useful metrics for determining which retweeted information is interesting.

3.1 Tweet and Retweet Properties

3.1.1 Retweet Groups

A Tweet has various attributes associated with it, which make up the features that describe that particular Tweet and its author. These properties relate the to the Tweet's content, its author, and other metadata, such as creation time, geographical location coordinates, language, and so on.
However, not all of these properties are necessary for use in this research and, as such, a particular Tweet, t, can have its relevant properties declared and be defined as follows;

$$t = (\text{text}, \text{count}_R, \text{author}_O, \text{author}_R, \text{orig})$$

Respectively, this represents the Tweet's text, its retweet count, and the *original* author of the Tweet. The final two values depend on whether t is a retweet or not and represent the author of the retweet and a reference to the original Tweet respectively and are nullified when t is not a retweet. Since a retweet is simply an extension of a class of Tweet, then the same properties can be assigned to retweets as to Tweets, except that in the case of retweets the values orig and author_R will be non-null.

A Twitter user, u, is represented by a Twitter account, and also has a set of properties. In relevance to the work in this thesis, these largely relate to the user's position in the social graph.
In particular, the set of followers and the set of friends of user $u \in G$ are denoted as $N^+(u)$ and $N^-(u)$ respectively, where;

$$N^+ = \{u_i \quad \forall \quad 0 \leq i \leq |G| \quad : \quad \exists \quad \overrightarrow{u_i u}\}$$
$$N^- = \{u_i \quad \forall \quad 0 \leq i \leq |G| \quad : \quad \exists \quad \overleftarrow{u_i u}\}$$

The Tweet attributes .author_O and .author_R are types of user. For example, since users represent vertices on the social graph, the author of a Tweet, t, has a follower count of $\deg^+(t.\text{author}_O)$ and a friend count of $\deg^-(t.\text{author}_O)$.

Since a Tweet can be retweeted more than once, the set of Tweets that are in the set of *all* Tweets, T, and are retweets of t is defined as;

$$RT(t) = \{r \in T : r.\text{orig} = t\}$$

Clearly, the retweet count of t is $t.\text{count}_R = |RT(t)|$.

Definition 3.1

A **retweet group**, denoted by $RG(t)$, describes the original Tweet, t, along with the set of the retweets of t, $RT(t)$. The term can be used to refer to the individual Tweet and retweets themselves or to the author users of those Tweets.

Retweet groups are useful for classifying a Tweet and the users who have retweeted it, and is appropriate when discussing the audience reach of a particular Tweet. Therefore, since t is also a member of this set, the size of t's retweet group is;

$$|RG(t)| = t.\text{count}_R + 1$$

Which can have a minimum size of two - the original author and at least one retweeter.

If $r_1, ..., r_n$ are the members of $RT(t)$ then the raw audience size of the group can be calculated thus (assuming $t.\text{count}_R \geq 1$);

$$\text{audience}(RG(t)) = \deg^+(t.\text{author}_O) + \sum_{i=1}^{t.\text{count}_R} \deg^+(r_i.\text{author}_O)$$

However, properties of Twitter's social structure dictate that this raw audience size is not an accurate calculation in most cases, as is discussed later in this chapter.

3.1.2 Retweet Trees

As a Tweet gains in popularity and attracts more and more retweets to be created from it, and since its retweets themselves can *also* be retweeted, then this ultimately results in the generation of a retweet *tree*, which represents the retweet group of a particular

Tweet. This tree is formed from the users who have retweeted the Tweet (or a retweet of the Tweet), and represents the original Tweeter and the various pathways taken by the Tweet as it is retweeted through the social graph.

The tree is not a representation of the actual social ties between the tree's nodes, as users are able to retweet Tweets and retweets sent from others that they do not follow. However, as is mentioned later in this chapter, most retweeting does generally occur between directly-linked users.

[**?**] also uses retweet trees to assist in illustrating information dissemination in Twitter, particularly in observing the Twitter reactions to the 2009 Air France airline crash.

The root of the tree representing every $RG(t) \forall t \in T$ is $t.\text{author}_O$ and, if t has been retweeted, each of the other nodes are

$$r_1.\text{author}_R, ..., r_n.\text{author}_R \; \forall \; 1 \leq n \leq t.\text{count}_R$$

A similar illustrative device is used by [**?**] in describing URL cascades in Twitter.

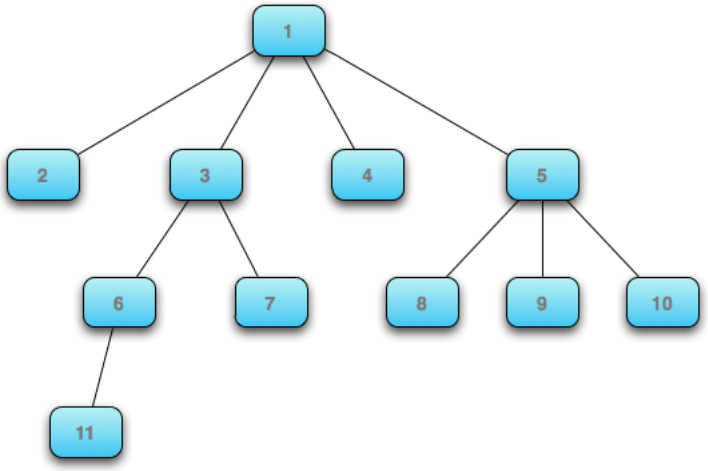


Figure 3.1: A hypothetical retweet pathway tree

Although these retweet pathways can technically be acyclic through the use of the manual retweet method, the case of a user retweeting a Tweet more than once is very rare and a user retweeting a retweet that they are already part of the upstream chain of

is even less likely to occur. The retweet button method simply does not support users retweeting a Tweet more than once.

As such, retweet trees are used in preference over retweet *graphs* as they help illustrate the temporal nature in terms of the order in which the retweets occur and the chains they produce.

3.1.3 Path-Length

In addition to retweet groups having a size property, a retweet groups's branch's *path-length* refers to the length of a particular retweet chain.

Definition 3.2

> The **path-length** of a single retweet chain in a retweet group is defined as the number of times a Tweet, t, is retweeted down the chain from $t.\text{author}_O$ to $r.\text{author}_R$, where $r \in RT(t)$ and where r is the leaf node of the branch of the retweet tree representing this chain.
>
> The **maximum path-length** of a retweet group is the largest path-length observed in the retweet group.

Figure 3.1 represents the users in the retweet group of a hypothetical Tweet.

This retweet group has a size of 11 and has 7 distinct retweet chains, the longest of which is the one traversing users 1, 3, 6 and 11.

The *maximum* path-length of this retweet group is therefore 3, as the leaf node of this branch is three hops away from the original author at the root.

As has been mentioned previously, when a user retweets a Tweet or retweet through the manual approach, it involves pre-pending the current state of the Tweet with the text `RT @<username>:`.

Therefore, a Tweet with the content;

`RT @user2: RT @user1: This is the body of the Tweet`

was originally authored by `user1`, then retweeted by `user2`, and then finally retweeted

by the author of this current retweet (a Tweet or retweet's author's username is not credited in the body of the text in this way).

It should be noted that this phenomenon can only be observed through retweets by the manual approach, since the button method always simply credits the original author, and not any of the internal members of the retweet group.

Although most retweets today are carried out using the button method, the manual approach still remained popular at the time the research in this chapter was carried out. This allowed for making useful observations of retweet patterns that could not be as successful later on.

3.2 More on Information Retrieval

A Twitter user electing to follow another user cannot, in most cases, predict precisely what the new friend will Tweet about in the future. The user has some *expectation* of the type of information they are likely to receive based on the previous Tweets of the new friend, which is generally the main cue the user can use to base the follow decision on.

Part of the follow decision is based on the notion of relevance judgement, which is an idea discussed at more length by [**?**] and is partly made up of the goal of achieving *affective stimulation* through *hedonic* searching as opposed to the use of *epistemic* searching.

3.2.1 Epistemic Search

An epistemic information search is one that involves carrying out a search with the purpose of finding out information on a particular topic (or set of) to satisfy a *desire for knowledge* [**?**], yet without an actual aim to solve any particular problem.

An example of this type of search is a 'crawl' through Wikipedia, in which a searcher may start at one particular page of interest and then follow links within that page to other related pages of interest that stem away from the source topic. In this case, the search 'parameter' is simply the name or title of the article the searcher wants to view. As mentioned previously, a followship between users is effectively a search parameter in Twitter, since the following user has elected to follow the new friend to receive information from him/her. It is clear that this type of 'searching' cannot be epistemic as the following user cannot know exactly the *type* of information they are going to receive.

3.2.2 Hedonic Search and Affective Stimulation

Hedonic searching is similar to epistemic searching in that it is also not carried out with the aim to solve an immediate problem, but is different in that it is done to search for fun or 'affective stimulation' [**?**].

A person can be said to be affectively stimulated if they view a piece of information that has some effect on the person, such as something that conveys emotion, something that is of particular interest to the person, or something that is capable of provoking some further thought.

With hedonic searching, users are not aware of the information that they are going to receive prior to searching and thus cannot really predict any level of affective stimulation.

This aligns more with Twitter usage, since users receive information that they cannot accurately predict. Any Tweets received that do provide interesting information can convey affective stimulation to the user. This is the type of information that becomes harder to identify amongst lots of noise, yet is also the type of information a user is more likely to retweet.

3.2.3 The Recognition Heuristic

A further metric for measuring information relevance in information retrieval is the recognition heuristic.
The recognition heuristic takes advantage of a person's memory and declares that if a person is able to recognise only one of two (or more) items, then he/she is more likely to judge the recognised item to be 'greater' or more important [**?**] [**?**].

Relating this to information received on Twitter, [**?**] found that a user recognising a Tweet's author significantly increases the chance that the user will decide to read the Tweet. Since a user must read a Tweet in order to make a decision on whether, or not, to retweet it, then the recognition heuristic transiently plays a part in a user's retweet decision also.

The authors also find that information about the Tweet itself, such as its text content and its retweet count, has much more of an effect on a user's read decision than information about the author, such as the followers count or Tweet rate. This also contributes to the declaration that information interest goes beyond the features surrounding a particular user and that user influence does not dictate interestingness of information.

3.3 Twitter Propagation Analysis

Understanding information propagation in Twitter is the key to also understanding how interesting information might be detected. Whilst it is known that the retweet count of a Tweet cannot be used alone in inferring interestingness, since this is simply a level of popularity tied in with the author user's influence, it is still a factor in that users are more likely to retweet interesting information than noise.

Of particular interest is to achieve an overview of propagation behaviours in Twitter; the patterns in the properties of retweet groups, such as their sizes and penetration

depth, temporal aspects of retweets and information on the social structure of Twitter itself with regards to propagation within it.

The remainder of this chapter involves an exploratory study of the retweet characteristics in Twitter to provide a further background, and which demonstrates the area's relevance towards the goal of inferring interesting information.

3.4 Retweet and Retweet Group Analysis

To assist in providing a further grounding in this area of research, a series of analyses were carried out into retweets and retweet groups. This section describes the processes and purposes of the analyses.

3.4.1 Data Collection Methodology

The analyses involve the examination of Tweets extracted from Twitter. Twitter's REST API v1 was used between 26th January and 24th May to collect around 26,000 Tweets, which represent a total of around 4,400 retweet groups. The complete set is made up of three subsets, the use of each individually is described later.
The relatively limited size of the dataset is acknowledged, yet it should be emphasised that these analyses are simply exploratory and are not used to answer or solve any specific problem.

The data collection involved a mixture of using Twitter's timelines and its search capabilities. Version 1 of the REST API supported retrieval of Tweets, 20 at a time, from the Twitter *public* timeline. Historically, this timeline contained the 20 most recent Tweets published by all the authors that have non-protected Twitter accounts, and it used to be visible on their website's homepage[1] to non-logged-in users.

[1] http://twitter.com

In particular, the public timeline endpoint was queried periodically to retrieve the current set of the most recent public Tweets. From all of the retrieved Tweets, the Tweets that were retweets were filtered out and stored.
Retweets, as mentioned earlier, are distinguishable since they start with the characters 'RT' followed by a username. It should be noted that when retrieving Tweets from Twitter's APIs that even retweets that were created using the button method begin with the same character sequence, allowing detection of these also.

Following storage, the content of the retweets were parsed in order to extract the text that the original Tweet contained. Sometimes, retweets using the manual approach are used to provide additional annotation to the Tweet. Although this can often be distinguished by the fact that the original Tweet content is inside quotation marks (" "), it is not true in all cases, meaning that sometimes the original text could not be reliably extracted programmatically by a machine.
In these cases additional queries were made to Twitter's search API in an attempt to resolve the problem, yet, failing that, the retweet was discarded.

Once the original text had been successfully extracted, this was used along with other metadata as query parameters to Twitter's search API in order to try and find the original Tweet and any other retweets of this Tweet. The search API uses approximate (or 'fuzzy') string matching, but quotation marks can be used to retrieve search results based on an exact string pattern[2].

Once the API search was complete (in some cases, with Tweets achieving many retweets, many API calls were required in order to page through results), the original Tweet could easily be identified as the only one of the set *not* starting with the sequence "RT". This provided a retweet group comprising the original Tweet and all available retweets of this Tweet.
On some occasions, more than one Tweet were each identified as the original Tweet and so the entire set was discarded. This could occur, for example, if many users may

[2]https://dev.twitter.com/docs/using-search

Tweet exactly the same text if it comes external sources, such as a news webpage, and means that the entire set of retrieved Tweets are not likely to be part of the same retweet group. In cases where no results were returned, the retweet was discarded and assumed an orphan retweet (perhaps as a result of a retweet of a Tweet posted by a protected Twitter account). And in cases where no original Tweet could be identified, it was sometimes possible to calculate it through cross-matching against other retweets in the retrieved retweet group, but if not they, too, were discarded.

The retweet groups were finally stored along with relevant metadata in order to carry out the studies described in the following sections.

3.4.2 Exploring Retweet Group Path-Lengths

The path-lengths of each chain in a retweet group can be calculated by identifying the users involved in retweet activity down that chain; from the original author to the final retweeter. The *maximum* path-length of a particular retweet group is the longest path-length observed in the group.

Identification of path-lengths can be carried out through parsing the text of a retweet, and following the citations. Although it cannot be guaranteed that all users will be properly cited in a chain, and there is no realistic method to verify this, it is felt that correct citations will be made enough times to make these cases relatively insignificant.

On average, the maximum path-length observed across the retweet groups was around 1.8, with the vast majority of retweet chains being between one and two edges in length. When one considers that many retweets are made through the button method, which removes citations of internal users in the chain and simply credits the original author and would therefore produce many single-length retweet chains, this average could theoretically be an underestimate.
[**?**]'s similar obsevations in the area also indicate a large number of groups with maximum path-lengths of one and two.

The longest observed maximum path-length was nine, which is a huge depth of penetration through the social structure since the total number of users involved in propagating the Tweet was ten. This, combined with the knowledge that social networks can represent a 'closer' social graph than the real world's six degrees of separation (see Introduction), shows how retweeting can have a huge impact in information spread amongst millions of people worldwide very quickly.

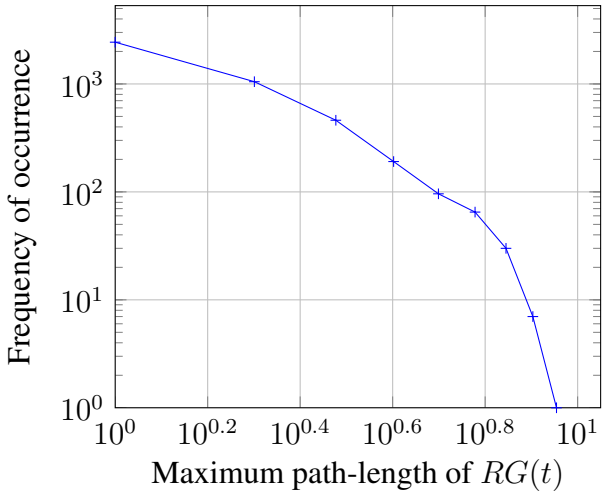


Figure 3.2: Log/log distribution of maximum path-lengths observed in $RG(t)\forall t \in T$', where T' is the set of Tweets analysed.

Also of interest is the relationship in terms of the social ties between different the different user members of a retweet group.

In cases where a retweet group's maximum path-length is precisely one, i.e. the situation where a user (or set of) has retweeted a particular Tweet only once, the retweeters at the leaves of this group's retweet tree follow the original author around 90% of the time.

This implies, therefore, that in the remaining 10% of cases, a retweeter has retweeted a Tweet from outside of their home timeline and has instead seen a Tweet whilst browsing through another user, who isn't a friend, timeline that the retweeter regards as sufficiently interesting.

This helps to demonstrate that the more followers a particular user has, the greater the

chance that another user somewhere has of viewing the user's Tweets and then having the opportunity to retweet them. The fact that 90% of retweets of a particular user are created by direct followers reinforces this further.

This particular property could also be partly due to use of the button method of retweeting, which does not cite intermediate retweeters, and thus always imply that the final retweeter directly retweeted the Tweet from the original author. However, there may, in fact, have been other retweeters in between the final retweeters and original author, each of which following the immediately upstream retweeter.

As such, this 90% follow probability between the retweeter and source user in 1-hop retweet chains is also likely to be an underestimate.

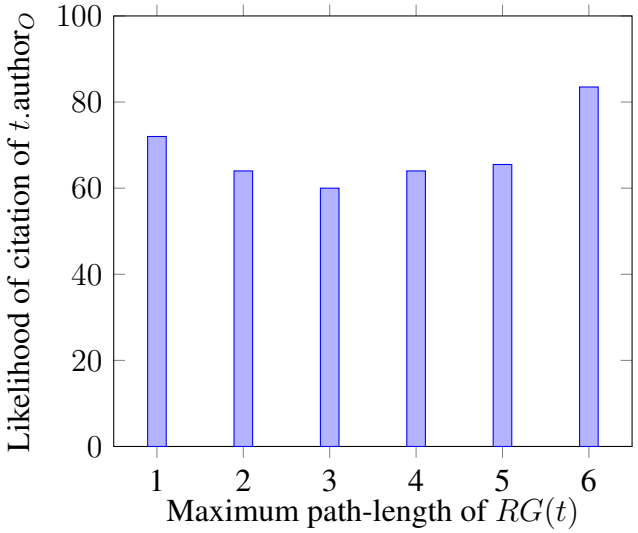


Figure 3.3: Proportion of cases where the original author is cited with varying maximum path-length of retweet group.

Further to this, in situations in which the maximum path-length of a retweet group is *greater* than one, retweeting authors in the group follow the author of the original Tweet about 40% of the time. It is clear from Figure 3.5 that retweet groups with a longer maximum path-length tend to have a larger size themselves. This increases the likelihood that the Tweet has been able to spread both further around the original Tweet's author's community, and also the potential for the Tweet to 'travel' to other

communities.

Since users from outside the source user's community are less likely to follow the source user, this explains the reduction in the followship likelihood between further downstream retweeters in the retweet chains and the original author.

3.4.3 Size of Retweet Groups

The distribution of $|RG(t)|$ across all of the original Tweets $t \in T$ collected from Twitter was found to follow a power-law type distribution, with a relatively large p-value of around 0.87. Figure 3.4 represents the complementary distribution function demonstrating the changing probability of a randomly generated X being greater than or equal to x, the 'current' value of $|RG(t)|$, at each stage.

The techniques used in this analysis are adapted from the methods and code provided by [**?**].

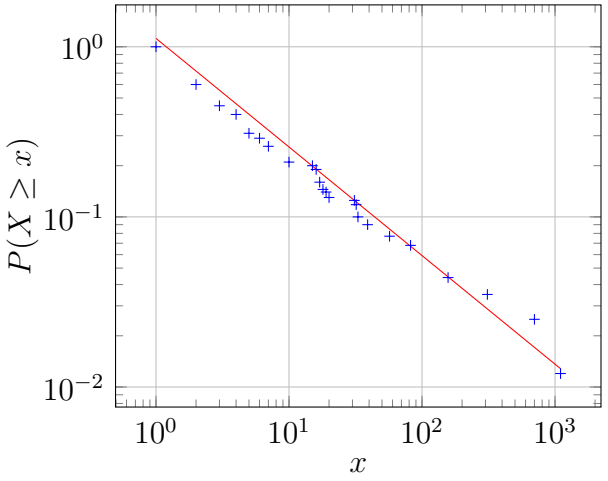


Figure 3.4: Maximum likelihood power-law fit for the cumulative distribution of retweet group sizes.

The mean group size from this dataset was found to be just below six, and the largest size was 284. The smallest $|RG(t)|$ were the cases in which $t.\text{count}_R = 1$, and which were significantly the most common occurrences.

Of interest also is the relationship between a group's size and its maximum path-length. Generally, the maximum path-length of a group, $RG(t)$, increases with $|RG(t)|$, indicating a mostly uniform growth in the retweet trees representing these groups - as might be expected. Thus this illustrates that as the retweet count of t increases, then the longer the retweet chains in $RG(t)$ are likely become. This would increase its penetrative dissemination away from the source and further facilitate its spread between communities, increasing its potential *audience size*.

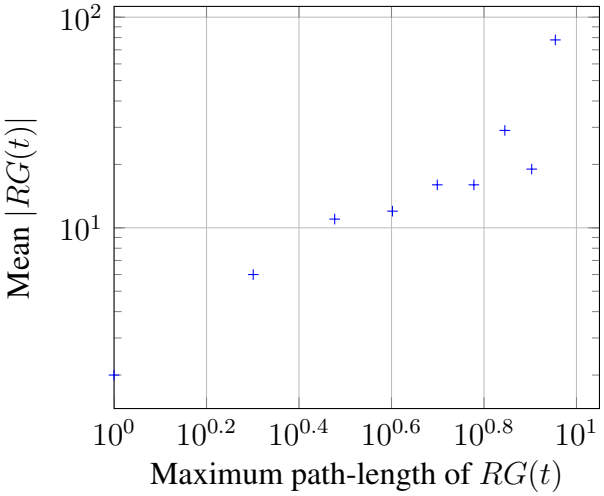


Figure 3.5: Log/log relationship between the maximum path-length and size of a retweet group.

3.4.4 A Tweet's Audience - How Many Users Can be Reached?

$RG(t)$'s (immediate) audience size refers to the number of Twitter users that have received t, either in its original form or as a retweet, r, such that $r.\text{orig} = t$, onto their home timelines. The term 'immediate' is used to signify the distinction between those users who passively receive the Tweet, due to following the original author or a retweeter, and those who see the Tweet whilst actively browsing through other user timelines or the public timeline.

Users in the latter group are therefore not direct followers of $t.\text{author}_O$ or $r.\text{author}_R \forall r \in$

$RT(t)$ and thus cannot be tracked as members of t's audience, which, as discussed earlier, can have its size calculated through the summation of the followers of the original author and each retweeter of t.

However, this audience calculation is naïve in that, particularly in the case of more tightly-knit communities, users who are authors of t or $r \in RT(t)$ are likely to share a subset of each of their followers. The more dense the communities, the more followers are likely to be shared between the authors in $RG(t)$ and, as such, the aforementioned audience size calculation is likely to be an overestimate in nearly all cases.
The following analyses of retweet group audience sizes relies on a dataset which began collecting at a later date than the general set used in this chapter, and thus the data represented in the rest of this section contains 2860 of the total 4400 groups originally collected. The longest maximum path-length of retweet groups observed in this subset was eight.

The *overhead* of a group, $RG(t)$, which attempts to address this problem, is related to the redundency in the audience and thus represents the number of cases in which a user receives a retweet that they have already previously received the original Tweet or retweet thereof - i.e. the number of times users receive a Tweet they've already seen. The overhead also takes into account users who might receive versions of the same Tweet many times.

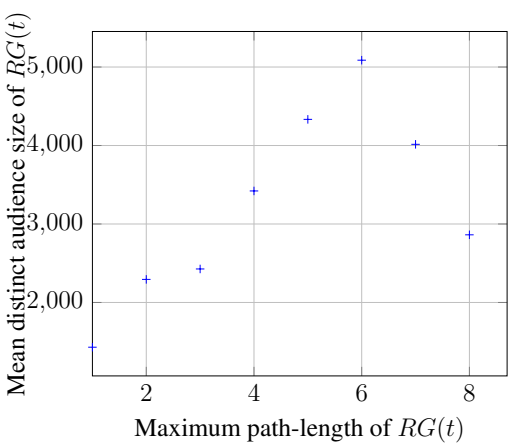
This overhead was found to exist in 71% of all observed retweet groups, further reinforcing that retweets often occur within communities containing users sharing links with other users. The *proportionate* overhead is the ratio of the overhead to the *distinct* audience size, which is the *absolute* number of users who have recieved the Tweet (or a retweet of) at least once to their home timeline. It should be noted that the audience size does not signify the number of users who have *read* the Tweet - rather the number of users who have the *opportunity* to read it.

Effectively, therefore the distinct audience size of a Tweet can be found by modifying

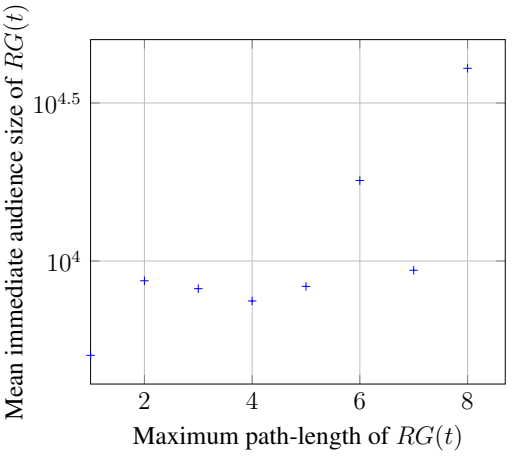
the earlier calculation:

$$\text{distinct audience}(RG(t)) = \deg^+(t.\text{author}_O) + \sum_{i=1}^{t.\text{count}_R} \deg^+(r_i.\text{author}_O) - \text{overhead}(RG(t))$$

Where the overhead of the group is simply the magnitude of shared followers of all authors in the group.



(a) Varying $RG(t)$'s *distinct* audience size with its longest path-length



(b) Varying $RG(t)$'s *immediate* audience size with its longest path-length

Figure 3.6: Comparison of the relationships between $RG(t)$'s distinct and immediate audience size and its maximum path-length $\forall t \in T$', where T' is the set of analysed Tweets.

Figure 3.6a illustrates, initially, that which might be expected; that the distinct audience size of a Tweet, t, is mostly proportional to the maximum path length of $RG(t)$. However, as the maximum path-length of retweet groups exceeds 5, then a *decline* in the distinct audience size is observed. This particular behaviour has an unclear cause, but it is felt that this could be to do with a saturation in the proportionate overhead's ratio at this stage - in particular, that retweet groups attracting many retweets are circulated more within communities than outside and between communuties.

At this stage, the overhead becomes so large, causing this reduction in audience size. This is significant in that the distribution of the non-distict over the increasing path-lengths demonstrates, mostly, a continuous positive correlation.

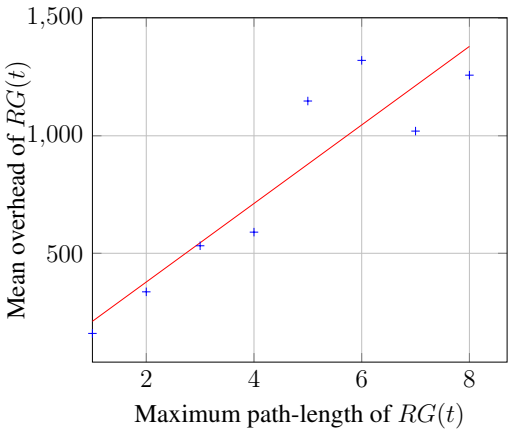
Three of the largest five overheads in the set occur in retweet groups which have a maximum path-length of one. The *largest* overhead was of a size over six times greater than the group's distinct audience size itself, indicating a massive overlap between the followers of the author of the original Tweet and the authors of its retweets. Whilst the audience overhead was only found to be greater than the distinct audience size in around 3% of observed retweet groups, it is still clear that the potential for overlap in the followers of retweet group members can be very large in more closely-knit communities - i.e. those groups whose representative trees are wide and shallow.
Groups having trees with longer path-lengths typically have a proportinately lower overhead, and the chance of achieving zero overhead increases as the retweet group size decreases.

The power of the retweet phenomeon in terms of how it affects the potential audience reach of a particular Tweet is discussed in further detail by the authors of [**?**], in which they find that a retweeted Tweet of sufficient interest can reach a very large number of users even if the original author has only a few followers. The same paper more specifically mentions that the audience size of a retweeted Tweet reaches, on average, at least 1,000 users, no matter the number of followers of the original author.
This is also clear in the results in this thesis, in that even Tweets with a short maximum path-length can still have a relatively large audience size.

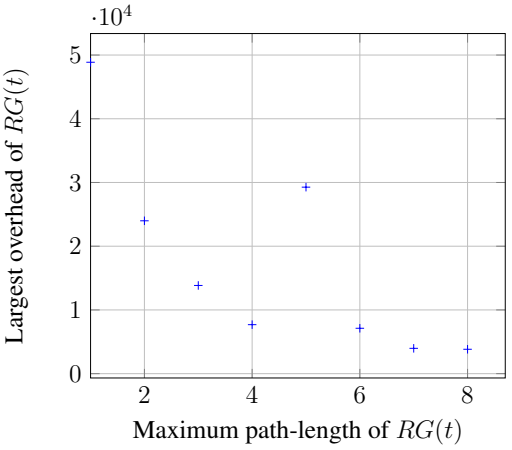
3.4.5 Retweet Groups on the Social Graph

Now that an understanding has been achieved in the behaviours and properties of retweets and retweet groups, it is important that the social ties between users in groups is studied. This will provide a grounding for the research in the following chapter, in which the social structure and its role in facilitating propagation, are discussed in more detail.

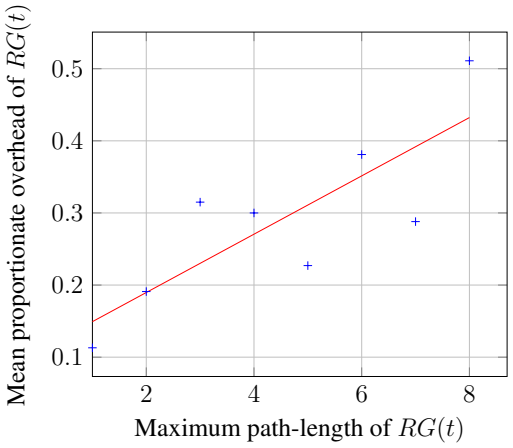
It has already been mentioned that the probability of a retweeting author following the



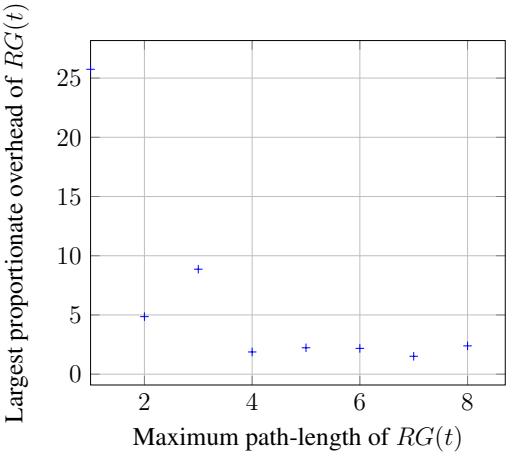
(a) Varying mean overhead with maximum path-length



(b) Varying *largest* observed overhead with maximum path-length



(c) Varying mean overhead *proportion* with maximum path-length



(d) Varying *largest* observed overhead *proportion* with maximum path-length

Figure 3.7: Relationships between $RG(t)$'s audience overhead properties and its maximum path-length $\forall t \in T$', where T' is the set of analysed Tweets.

original author in single-length retweet chains was found to be around 90%. However, in retweet groups with longer chains, a decrease in the likelihood of the final retweeter (the user at the bottom of the retweet tree) following the original author was observed. Indeed, on average across all retweet groups, the final retweeter follows the *previous* retweeter in a particular retweet chain in around 67% of cases.

The final retweeter is defined as the author of the chronologically final retweet in the

group, and is not necessarily the user at the leaf node of the group's longest retweet chain.

It is interesting that this value should be about 20% lower than in single-length maximum path-length groups, and it suggests that users have a greater chance of 'stumbling over' retweets found on non-friends' timelines whilst browsing through other users. Since it has been shown that with an increase in maximum path-length an increase in the audience size is also observed, then this demonstrates the increased chance of discovery of the Tweet through users searching through others' profiles.
In cases where the maximum path-length of $RG(t)$ is equal to one, then the audience size is far smaller and thus there is a lower chance of users who aren't followers of $t.\text{author}_O$ or $r.\text{author}_R \forall r \in RT(t)$ finding the Tweet.

In addition, there is some evidence of user influence playing a role in the analyses of these data. In particular, in the 67% of retweet groups in which the final retweeter *does* follow the author of the retweet (or original Tweet) directly 'upstream', the latter user has, on average, around 950 followers. Inversely, in the remaining 33% of groups (in which the author of the final retweet does *not* follow the preceding author), the preceding author has an average of 600 followers, thus implying a significant difference in the retweet potential with varying author influence levels.

This is further accentuated when one studies the follower connections of $t.\text{author}_O$. Whilst it was found earlier that the likelihood of a $r.\text{author}_R$ following $t.\text{author}_O$ when the maximum path-length of $RG(t)$ is greater than one is around 40%, the average follower count of $t.\text{author}_O$ has a four-fold increase (from about 550 to 2,000) when he/she is also followed by the final retweeter. In fact, in groups of all maximum path-lengths, $t.\text{author}_O$ had a consistently higher follower count when followed also by the final retweeter of $RG(t)$ than when not followed.
This particular behaviour also helps illustrate that a user is more likely to be retweeted when having more followers - in this case, having four times the follower count increases the correlation dramatically (40% to 90%). The follower count can, therefore,

be directly related in this way to the discussions of user influence in [**?**], and also of users using retweeted Tweets to passively 'advertise' themselves.

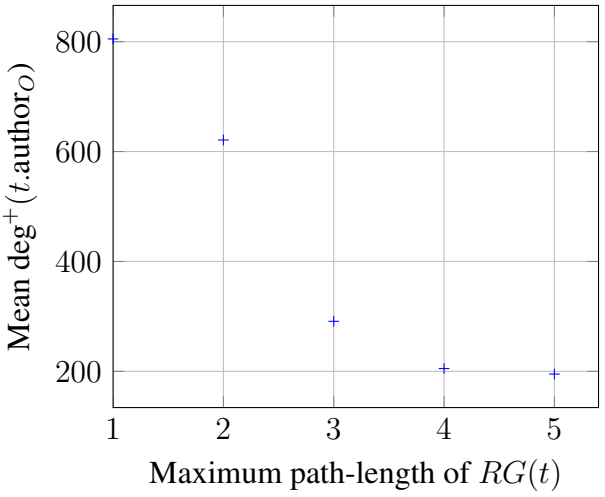


Figure 3.8: Analysis of variance in deg$^+(t.\textbf{author}_O)$ as $RG(t)$'s maximum path-length increases.

Strangely, it was found that $t.\text{author}_O$'s follower count actually diminishes with increasing maximum path-length of $RG(t)$, indicating further penetrative depth of propagation when the original author has *fewer* followers. The collected retweet groups that contained longer retweet chains often also contained retweet chains that were much shorter. For example, a group containing chains with path-lengths of five, or more, are also likely to contain many more chains with path-lengths of one and two (as is implied in the distribution in Figure 3.2).

There are, therefore, various possible explanations for this property, including the argument that users with many followers are generally likely to be part of a large community of users, from which retweets are not transmitted. Users that are part of several communities, and are therefore less involved with any given one, may find that their Tweets have the potential to be retweeted a further distance.

Additionally, and more interestingly, it is possible that users possess some awareness of their local networks and the users within them. A user, who is part of a large community with lots of obvious follower overlaps occurring between the members, may

decide *not* to retweet a particular Tweet if he/she feels that many of their own followers may have already seen the Tweet due to them also having a high chance of following the original author.

A final analysis on the social ties between users in retweet chains is carried out on the followship pattern of authors throughout the chain. Let h be the number of hops (or edges in the retweet tree) between two users in a retweet chain. It was illustrated in earlier sections that, when $h = 1$, the likelihood of the later retweeting author following the upstream author is around 67%.
However, as h is increased, then the followship likelihood mostly consistently decreases (see Figure 3.9), as might be expected. This illustrates how longer retweet chains do indeed increase both the likelihood of the Tweet reaching further through the social structure and the chance of achieving a smaller proportionate overhead.

Further to this, of the 67% of retweeters who *do* follow an upstream author at $h = 1$, only 19% follow also the upstream author at $h = 2$. In these cases, the latter has an observed average of around 3,000 followers.
In the 81% of cases when the user at $h = 2$ *isn't* also followed, then the upstream author has a much lower average follower count of about 520.

It is, therefore, sensible to assume from these analyses that Tweets are forwarded more between groups of less-connected users, highlighting the notions of social network awareness and of community-hopping. If retweets were usually circulated around more closely-knit communities of users, then the followship likelihoods would be generally greater, more uniform, and consistent throughout the retweet chain. Users would have as much of a chance of following their immediate upstream neighbour author in the retweet chain as they would an author further upstream.

As mentioned near the start of this chapter, the author of the original Tweet should be cited by the `RT @<username>` sequence observed *closest* to the retweet body, where the `<username>` is the username of the original author user. Rather than specifically looking for the author's Tweet appearing in this location, Tweets were examined to

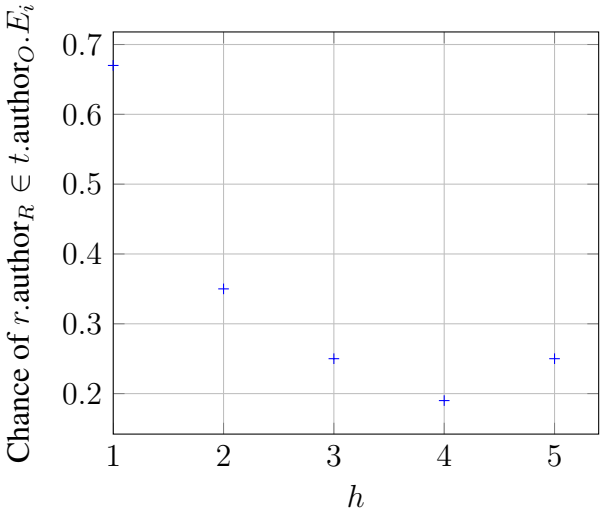


Figure 3.9: Relationship between the followship chance of r.author$_R$ (where $r \in RT(t)$) to t.author$_O$ and the increase in 'distance' between r and t given by h.

check for the existence of the author's username being mentioned *anywhere* in the Tweet content, and was found to exist in about 68% of Tweets.
This frequency did not vary with any consistent correlation upon changes to the maximum path-length or retweet group size, and so it is assumed that users do feel the need to credit the original author more so than not.

3.4.6 The Temporal Properties of Retweets

The final set of analyses in this chapter relate to time's influence on retweet propagation. This provides insights into how quickly information can spread and, when combined with the knowledge of the social structure and audience, how this can relate to the rate of information dissemination and consumption.

A general observation is that the elapsement of time between the original Tweet and the final retweet of retweet groups of varying maximum path-lengths increases with path-length, indicating that if there are more hops for a Tweet to travel down between users then it takes longer to do so. However, this correlation is only really applicable

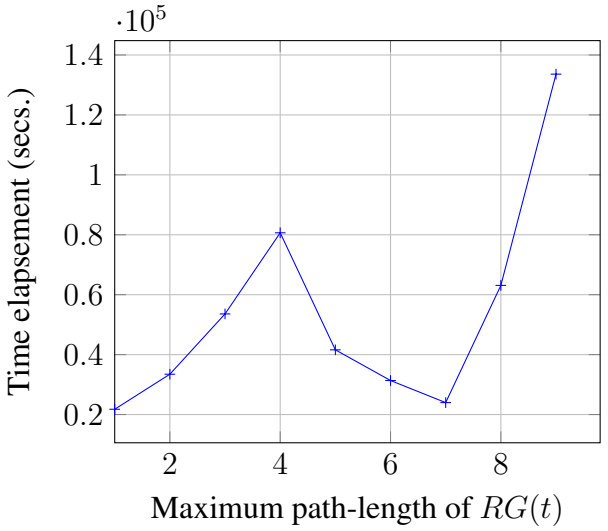


Figure 3.10: Comparison between the average time elapsement from t to the final $r \in RT(t)$ and the maximum path-length of $RG(t) \forall t \in T$‘, where T‘ is the set of analysed Tweets.

for shorter retweet chains and, particularly in cases where the maximum path-length is five and above, this pattern is not consistent.
The groups with shorter maximum path-lengths more uniformly increase in temporal elapsement with increases in maximum path-length in a linear fashion roughly proportional to $v = \frac{s}{t}$, where the distance, s, is the hypothetical distance given by the number of hops between users, thus indicating that the speed, v, of propagation remains relatively constant.

Despite this, there are conflicting arguments for patterns observed in retweet propagation speed, which rely on various different factors.
As mentioned, the time taken for a Tweet to reach a specific path-length could be a function of the path-length itself, where as the path-length increases, then so does the time taken for the Tweet to be retweeted to the end of the chain.
Inversely, Tweets that are especially popular, possibly as a result of being particularly topical (such as in the disaster cases mentioned in the Introduction), may be retweeted more quickly by users so that the information is spread more quickly. In these cases,

retweet groups with longer retweet chains may complete their trees more quickly than those groups with much shallower retweet trees.
Similarly, user influence could play a role in dissemination speed; if a Tweet is retweeted by a user with many followers, then there is an increased likelihood of propagation through this user. Whilst this could, in addition to the previous argument, cause longer retweet trees to be completed more quickly than groups with shorter trees, it could also facilitate 'faster branches', in which particular long branches grow faster and reach their leaves more quickly than shorter ones in the same retweet tree if the other branches consist of less-influential author users.

There is not enough evidence provided in this analysis to make any inferences towards a generic pattern of retweet group growth speed, and it is believed that this growth is governed by many more factors than the Tweet itself or the social structure alone. As such, there is no predefined rule for predicting the spread of dissemination in this way, since the retweet path is an unknown feature, with too many variables and conflicting arguments.

The temporality of retweets has been the focus of some researchers, including the authors of [**?**], who also used retweet trees as an illustration of the propagation pattern produced by Tweets. They found that, generally, half of all retweet action on a Tweet occurs *within an hour* of the Tweet being posted, and that by the end of the first day, 75% of all retweets of the Tweet will have been carried out.
The authors also conducted an analysis on the elapsed time of a Tweet's travel between hops as it is retweeted. Although they observe a flatter time elapsement initially, indicating that Tweets travelling over the first few hops are retweeted almost concurrently, they also found there to be a general incline in time elapsement over the shorter path-lengths. After this point, as seen and discussed in the analysis earlier, the elapsement becomes more 'noisy'.

An interesting notion that is not directly addressed in this thesis is that the time a particular Tweet is authored may have some effect on its propagation speed. Just as

'prime-time' television achieves higher audience ratings as it is at a time of the day when many people are at home and relaxing, Twitter may also have a time window in which its users are more active. For example, if a user posts a Tweet at a time when many of his/her followers are asleep, then the immediate audience size of the Tweet can be significantly reduced.
If there are fewer initial users viewing the Tweet, then the likelihood of retweet, as a function of this, is also reduced. This could have an effect on the perceived popularity of the Tweet, although since, by definition, there are fewer active users on Twitter at this time, then the number of Tweets sent during this period will be much smaller. We therefore do not take this factor into account during our experimentation in later chapters.

3.5 Summary

In this chapter, a set of initial exploratory analyses have been undertaken into the behaviour of retweets and retweet activity in Twitter, the properties of retweet groups, the relationships between the propagation graph and the social graph, and briefly into the effects of time on Tweet dissemination.

The analyses were found to support and complement the findings of other research in the area, including the notions of message cascading [**?**] and the relationships of this to the interconnection of users on the social graph through communities [**?**].
Trees representing the retweet groups were found to grow in a variety of ways, from thost illustrating long retweet chains, indicating a high level of inter-community dissemination, to shorter and wider trees, in which propagation can still be widespread but not as likely to disseminate to other communities.

User influence, in terms of an author's follower count, was observed as being an important factor in facilitating information spread, implying that popular users also produce popular information, since these users are more likely to achieve more retweets.

These inferences have helped to describe the multi-dimensional principles of retweet groups in terms of the features governing their spread over the social graph, and the quickness with which many users can be exposed to a Tweet. Although it is important to have an understanding of user psychology, and the thought processes behind the retweet decision, of most interest in this chapter is the analysis of the social structure.

3.6 Taking the Investigative Research Further

Twitter's social structure has been found to have a large effect on Tweet propagation, since it combines the features observed around user influence (in the naïve form of a user's follower count) with that of communities and sub-graphs of dense and sparse user interconnections.

In the following chapter, interests are focused on the topiological structure of user followships by investigating further into the flow of information between users as they are arranged in different ways in order to develop a method to infer information interestingness taking into account these information flow properties and user influence. It is clear that different Tweets can have a different level of *quality* in that Tweets that are retweeted have a greater chance of being interesting, but does the way in which the social structure of users is formed also have a quality in terms of supporting retweeting?

Chapter 4

Analysis of Twitter's Social Structure

In the previous chapter, a series of studies were conducted into Twitter with respect to message propagation through retweeting. In particular, research was done to provide an understanding of the patterns produced through retweets and how their properties relate to the Twitter users that the Tweets 'pass through'.

Of particular interest, however, is the social graph underlying Twitter, which describes how the users are interconnected and which dictates the information flow between them. It has been discussed that users with a higher follower count are more likely to have their Tweets retweeted, due to there being more users available to *see* the Tweet, and that some users can have their Tweets forwarded through many hops indeed, so that information may be passed between different communities of users.

In addition to the effects of user influence, several other factors also govern an individual retweet decision of a given user for a particular Tweet. These include properties of the Tweet, such as whether, or not, the Tweet contains a URL, whether it mentions a particular user, whether the user even has an opportunity to *view* the Tweet, and so on. These factors account for the individual user's retweet decision and the almagamation of every user's retweet decision on the Tweet describes the Tweet's overall retweetability, which essentially determines how far the Tweet can propagate.

However, it is believed that the topology of the network, below the level of user influence and other factors, can play an important role in facilitating (or inhibiting) Tweet propagation by opening and closing available retweet pathways between users and

groups of users.

Whilst retweet decisions based on Tweet features alone, such as the actual content of the Tweet or the contents of a document a URL in the Tweet points to, may imply a level of interest in the Tweet, the influence of users has a very large impact on how many retweets a certain Tweet receives. Thus, abstracting the concepts away from user influence may help in discovering methods for deducing which information is actually interesting.

Twitter's social structure has earlier been described as being built from users creating edges between themselves through the act of *following*. A followship defines the direction of travel of information from the follower to the friend, and this illustrates how users with many followers immediately have their Tweets made available to many more users before any retweeting even takes place.

As more edges are constructed between users the global initial spread of Tweets is increased, and, when the addition of retweets is considered, this has an larger effect. Although other intervening factors have been mentioned earlier, such as the notion of a user's network awareness and of user influence, the organisation of users on the graph and the differences in observed propagation pattern is an interesting route for research towards uncovering the properties surrounding interestingness.

In this chapter, various social network structures are constructed in order to simulate retweet behaviour between users on Twitter. The behaviours are studied with the aim to research the propagation patterns observed in different network structure types. Non-realistic and realistic graphs are built in order to highlight the low-level propagation characteristics in these networks and the similarities between more realistic simulated networks and Twitter's own social graph.

This research is then used to generate a methodology for estimating Tweet interestingness based on an *expected* Tweet popularity, as is discussed further later in the chapter.

4.1 Observing Differences in Propagation Patterns Between Different Network Structures

In the previous chapters it has been realised that certain Tweets with particular properties may imply a certain quality that affects a user's retweet decision on the Tweet. However, in addition to Tweet quality, of interest also is the potential presence of a graph 'quality', in that particular network structures may possess benefits for propagation, or at least have an effect on how Tweets are spread.

In this section, to help in addressing this research area, simulations are carried out in three different network topologies - a path (or 'linear') network, a random network, and a scale-free network. In the experiments, individual user *decisions* are used as the bases for demonstrating retweet behaviour.

The simulation algorithm and ideas behind the model used for generating the simulated users' retweet decisions are adapted from the work carried out in [**?**] and [**?**], which introduces methodologies for illustrating Tweet spread through a given network of users, and the simulations can be used to produce a retweet group for a given Tweet.
From the analyses of the simulation experiments, of interest is whether, and how, changing the network structure does affect retweet propagation patterns, and whether a simulation can mimic Twitter's own behaviour in terms of retweet spread.

Measuring retweet behaviour is carried out through studying the distribution of retweet group sizes that result from running the experiments, as is described in later sections.

4.1.1 Overview of the Simulation Algorithm

The algorithm covers the simulation of Tweet propagation through a given set of connected users by emulating retweet decisions of each user who receives the Tweet. The retweet decision is made using a prediction based on a logistic regression classifier, as is described below.

[**?**] developed a simulation algorithm which was found to be capable of accurately predicting retweet decisions using a logistic regression. These methods were modified and adapted to fit the purposes of the simulation algorithm used in this section.

In essence, the simulation initially requires a graph of connected users, U, and a Tweet, t, which will be introduced to the graph and retweeted between the users. It begins by initialising a set of users, S, to contain the followers of a particular $u_s \in U$, which represents the user $t.\text{author}_O$. As such, users in S form the set of users to have t or a retweet of t currently on their home timelines and available to retweet. In this case, t is used to denote both the original Tweet and any copies of it made through retweeting.

The procedure then iterates over timesteps, at each stage checking the retweet probability of each $u \in S$. If u's retweet probability is suffuciently great for t, then u retweets t by being removed from S and then added to $RT(t)$, which represents the set of users who have retweeted t. The followers of u are then added to S, since these users now also hold t and have the chance to make the retweet decision.

A threshold value, TH, is used to emulate the notion of the Tweet 'decay' experienced when one users a Twitter client or the web interface. The reasoning behind this is that as time goes by, more and more Tweets arrive onto the recipients' home timelines. This pushes the previous Tweets further down, whether they are interesting or not. Tweets may be ignored and not retweeted if the user has not viewed their home timeline for a while or if the user decides the Tweet is not of a sufficient quality to retweet it. If a Tweet is pushed down to the extent that is out of view, or out of the current page, then the chance of that user retweeting that Tweet is reduced. Thus, if a user is in S for more timestep iterations than specified by TH, then the user is removed from S, meaning that it can no longer have the chance to retweet the Tweet.

Users who have retweeted t, or are unable to do so (either by having previously retweeted it or by exceeding TH) are prohibited from being (re-)added to S.

The algorithm terminates either when the timesteps thus far iterated exceed the maximum allowed, T, or when S becomes empty. This results in the retweet group, $RG(t)$, which comprises the final set, $RT(t)$, along with the initial u_s. As in the previous

chapter, $t.\text{count}_R = |RT(t)|$.

Therfore, the additional necessary components to run the simulation are the facilities for building a user graph, a constructed Tweet, and functionality for generating a retweet probability for each user who receives the Tweet.

Algorithm 1 Simulation of retweet decisions on t in a given network of users, U

procedure SIMULATE(graph of users U, tweet t)
    $RT \leftarrow$ empty set ▷ To hold users who retweet t
    $T \leftarrow$ number of timesteps allowed
    $TH \leftarrow$ maximum timesteps ▷ Emulate t 'slipping down' timeline
    $us \leftarrow$ source User selected from U
    $S \leftarrow$ initialise to followers of us

    for all ti in range $(0, T)$ **do**
        for all $u \in S$ **do**
            $P \leftarrow$ retweet probability of u on t in range $(0, 1)$
            $r \leftarrow$ random number in range $(0, 1)$
            if $P > r$ **then**
                Remove u from S
                Add u to RT
                Add followers of u to S
            else
                Increment u.TIME_HELD
                if u.TIME_HELD $> TH$ **then**
                    Remove u from S ▷ u has held t for too long in timeline
                end if
            end if
        end for
        if $|S| = 0$ **then**
            Return RT ▷ No more users can retweet t
        end if
    end for
    Return RT
end procedure

4.1.2 Generating a User's Retweet Probability

As prevoiusly mentioned, [**?**] used a predictive model for retweet decisions based on a logistic regression, which was demonstrated to be capable of accurately predicting a user's retweet chance on a given Tweet. The regression was trained on a set of user, tweet and context features in order to classify a likelihood on the binary decision: retweet or no retweet, such that if $P = 1$ then the retweet will definitely occur.

Machine Learning

Machine learning is the term given to the family of techniques that allow a program to make predictions for the outcome of unseen instances based on an observed and known history of occurrences. There are many types of machine learning classifiers that are suitable for different purposes, such as for predicting an expected outcome from a set of nominal categories, for predicting a value from a continuous range, or for predicting the *probability* of a binary outcome.

Most machine learning techniques involve the training of a predictive model, which contains the information on known outcomes for a set of features. The model is then used to estimate an unknown outcome, usually with a probability on the *confidence* of the classification, for new sets of instances.

For example, consider three attribute variables, A, B, and C, each of which can be equal to one of two nominal values; TRUE or FALSE. A particular machine learning algorithm trains a model based on its knowledge that;

- $A \leftarrow$ TRUE, $B \leftarrow$ FALSE $\Longrightarrow C \leftarrow$ TRUE
- $A \leftarrow$ FALSE, $B \leftarrow$ FALSE $\Longrightarrow C \leftarrow$ FALSE

Although training of predictive models nearly always involves using more than two instances, the history of these example instances indicate that C is more strongly associated with A than with B. As more instances are added showing similar patterns,

then the association becomes stronger, to the extent that the classifier will predict $C \leftarrow$ TRUE in instances where $A \leftarrow$ TRUE (and vice versa) with higher confidence.

In this case, A, B, and C are known as the 'features', and a set of such features form the 'instance'. Once a trained model has been constructed, the machine learning algorithm will only be able to make predictions using instance features it has knowledge of. For example, if the example classifier was now given an instance containing a feature D, then it will not 'know' how changes in D will affect C's outcome.

If there is not a strong correlation between the features in a dataset, then the confidence of the classification of a particular feature will be weaker. Although this example has focussed on boolean (nominal) data types, many machine learning classifiers are able to work with features that are higher dimensional nominal values, contunuous reals, and so on, and will apply weights to the different features based on their level of influence over other features in the instance.

The Logistic Regression

Logistic regression analysis can be used as a machine learning classifier for working with binary outcomes based on a set of predictor variables (or features) [**?**], which makes it an appropriate approach for predicting a binary retweet decision. Logistic regressions have been frequently used in retweet analysis [**?**] [**?**] [**?**] [**?**] [**?**], as discussed in the Background chapter.

An implementation of the logistic regression algorithm was written in the Python programming language, which formed the basis of calculating the value for the retweet probability, P, based on a set of feautures of the Tweet and author user.

4.1.3 Summary of Training Features

[**?**] used the approach in order to accurately model retweet decisions in Twitter. A set of around 50 different features were used to train the logistic regression, with the retweet outcome (TRUE or FALSE) being the predicted classification in each case. These features included Tweet-related features (such as content analysis, inclusion of URLs, etc.), and network and user features (followships, mentions, etc.).

Since the network structures themselves, and the propagation *patterns*, are what are of interest in this section, the simulation is significantly simplified by using far fewer features, yet ones which are features that have been shown to have a strong influence on the retweet decision. As long as a consistent set of feature groups and values are used, the properties of the retweet groups observed should demonstrate the varying behaviours across the different user structures.

As such, each instance comprised the following four features associated with each Tweet, t, and where u is the user currently making the retweet decision, RETWEET;

Feature	Data type	Description
FOLLOWS	{TRUE, FALSE}	TRUE if $u \in N^+(t.\text{author}_O)$
FOLLOWED	{TRUE, FALSE}	TRUE if $u \in N^-(t.\text{author}_O)$
MENTIONED	{TRUE, FALSE}	TRUE if u is mentioned in t's content
URL	{TRUE, FALSE}	TRUE if `http://` or `https://` in t's content
RETWEET	{TRUE, FALSE}	TRUE if $u \in RT(t)$

Table 4.1: Training features for the logistic regression.

The URL feature has, in the literature, often been found as a large impacting feature on retweets in Twitter, especially in [**?**], who use it as their basis for determining .nd identifying interesting Tweets.

4.1.4 Training the Model

In order to train the logistic regression model, data was required from Twitter so that the sets of feature instances could be built.

Data collection for these experiments again utilised Twitter's REST API, which was queried between March and June 2012 to collect a set of around 12,000 Tweets and retweets. Since these dates were before the mandatory switch-over to v1.1 of the REST API, the public timeline could again be used to collect the data without the necessity of crawling through the social graph.
In this case, it was particularly necessary that non-retweets were also collected in order to provide the negative case when training the regression model and to ensure that there were instances where the RETWEET feature could be FALSE.

In cases where the collected Tweet was a retweet, further calls were made to the API to determine the relationships between the retweet's author and the original Tweet's author in order to satisfy the required FOLLOWS and FOLLOWED features. Where the collected Tweet was not an instance of retweet, there is no original author to examine the relationships between. In these cases, further Tweets were retrieved for the user in order to find their retweet rate in terms of the ratio of retweets to Tweets on their user timeline and an analysis of the relationship between these and the original authors. This was used in conjunction with the user's follower and friend count to determine a probability of the 'faux' followships. As mentioned, the accuracy of the retweet counts obtained through the simulations is not particularly important; of interest is the propagation patterns observed over the graph structures.

After storage, the regression model was trained using features extracted from the raw data, which the simulation algorithm could then use to generate the required retweet probability, P.

4.1.5 Running the Simulations

Once the model had been trained, the simulations could be run. In each simulation experiment, a network of users was generated, as described in the next section, and a Tweet object was created.
This Tweet object contained information on whether or not it contained a URL and if it mentioned one of the users in the generated network.

Various parameters - such as TH, the size of the user network, U to be generated, t's Tweet features, and any weightings on the decision probability prediction generator - could be altered to affect the strength or correlation of the patterns produced by the different network structure types in the simulations.

4.1.6 Network Analyses

In this section, three network structures are assessed in terms of the differences in the patterns of propagation each expresses. Each generated graph is *directed* in order to illustrate the followships between the user nodes, and to support the use of the FOLLOWS and FOLLOWED features required in the decision probability calculation.

In each case, the same set of generated Tweets were used, but different structures required the various parameters to be set slightly differently and, as such, each network structure will present with different proportionate retweet group size distributions when highlighting the different patterns.

Path Network

The first assessed structure was to illustrate the pattern on a non-realistic social network structure; a path network.

Path networks are one of the simplest type of graph, and a linear directional path network consists of the graph of users, U, of size n, in which each user $U_i \forall 0 \leq i < n$

is followed by user U_{i+1}. As a result, each $u \in U$ has precisely one follower and one friend, except the users U_n and U_1 respectively.
n is the only parameter necessary in the construction of this user graph.

Figure 4.1: Example of a path network.

In this graph, the size of the retweet group is, by definition, equal to the depth of penetration, as there is only one path (or retweet chain) available for propagation to occur down. As such, in each case, the retweet tree representing a resultant retweet group formed in this type of network will have the same structure as the graph itself, yet with a size dependent on the collective retweet decisions of the users.

Since each internal user has only one follower, the likelihood of a retweet decision being positive at each timestep is somewhat progressively reduced, and thus the retweet count is much more likely to tail off sooner than in graphs with more propagation avenues. This is also due to the fact that each retweet can only reach an audience of size 1 at each time step, and thus the 'survival' of the Tweet cannot rely on a summation of many users' retweet decisions.

The likelihood of a particular user achieving the opportunity to receive the Tweet, in order to then retweet it, becomes the product of the probability function the further it travels through the graph, in which user U_i requires each user from U_1 to U_{i-1} to first make a positive retweet decision. For example, if each user has probability p of retweeting the Tweet, then each user's chance of retweeting the Tweet is $\frac{1}{p^i}$, where i is the position of the user in the graph.

Therefore, as might be expected, the frequency distribution of retweet group sizes shows a half-life type behaviour demonstrating the logarithmic pattern with many small retweet groups followed by a series of exponentially smaller groups.

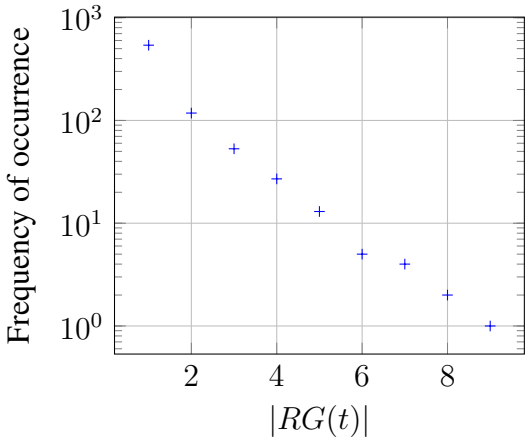


Figure 4.2: Frequency distribution of retweet group sizes in path network simulations.

This user structure illustrates well how some users that might find the Tweet interesting, and who may then decide to retweet it, do not even get the chance to view it in order to *make* that decision. Although this is accentuated in this structure, the same principle applies to any non-complete social graph, and demonstrates how the way users are connected can have a large impact on the retweetability of a particular Tweet.

Random Network

The random network was the next user structure to be analysed. Although it is certainly more similar to a real-life social graph than a path network, it is much more basic and uniform and does not consider user communities and clusters or different levels of influence in users in terms of differences in follower and friend counts.

A random social network is defined as the case in which the graph of user nodes, U, and where $n = |U|$, consists of each user, u, having probability p of following each other $u_i \in U \forall 0 \leq i \leq n$ and where $u_i \neq u$. Thus, as p is increased, then the likelihood of u following a u_i increases, causing the overall network edge density to increase. In general, therefore, the average number of followers and friends of a user is proportional to $p.n$.

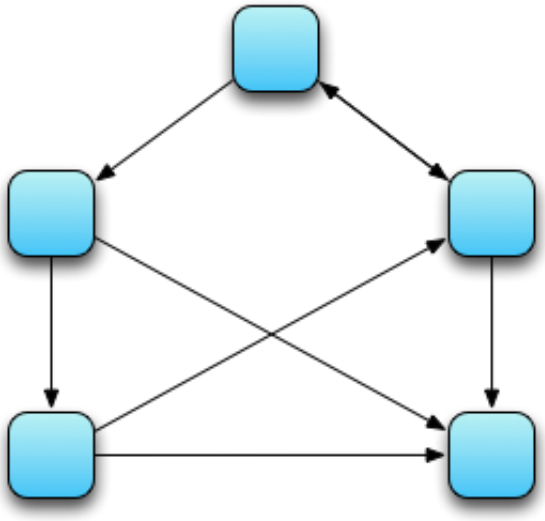


Figure 4.3: Example of a random network where $n = 5$ and $p \sim 0.5$.

The only parameters needed for constructing such a graph are n and p.

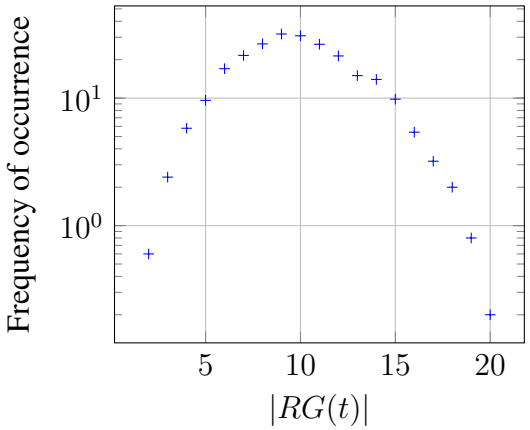


Figure 4.4: Frequency distribution of retweet group sizes in random network simulations.

The frequency distribution demonstrates a very large proportion of mid-range values for $|RG(t)|$, indicating that Tweets tend to have a consistent spread amongst the network, as might be expected. There are few smaller groups since there are no users that have disproportionately smaller spheres of influence, and each user has many incoming edges and a similar number of outgoing edges. As such, there are more mid-range retweet group sizes than smaller ones.

However, as in any distribution so far examined, the distribution of retweet group sizes must eventually tail off due to the natural eventual reduction in positive retweet decisions being successively made as retweet chains increase in length.

Scale-Free Network

The final network structure examined in this section is the scale-free network. Also known as 'small world', scale-free graphs are generally known to be representative of the general structure of 'real-life' and online social networks [**?**] and, indeed, they are also used to describe the interconnections of real-world properties, such as friendship groups and food webs [**?**] [**?**].
Essentially, scale-free networks dictate that there are a small number of nodes with a high degree and many nodes with a low degree, and are usually generated through some form of preferential attachment algorithm. Thus, this type of network has support for the consideration of user communities and influential users in terms of those demonstrating a disproportionately large follower count. The other user structures studied do not have the scope for emulating this property of inconsistent interconnection between the user nodes.

Scale-free networks are constructed such that the distribution of the degree of the graph's nodes follow a power-law in that the distribution of the number of vertex edges across the graph is logarithmic.
For these analyses, NetworkX[1], a Python graph and networking package, was used to generate directed scale-free graphs of users, which essentially accepts a network size, n, and edge density d as the graph construction parameters.

From simulations of the algorithm through these scale-free networks, a logarithmic trend is observed similar to that demonstrated from the 'real' Twitter data analysed in the previous chapter and published in [**?**], and the similarities in the distribution pattern is illustrated by Figure 4.5.

[1] http://networkx.lanl.gov

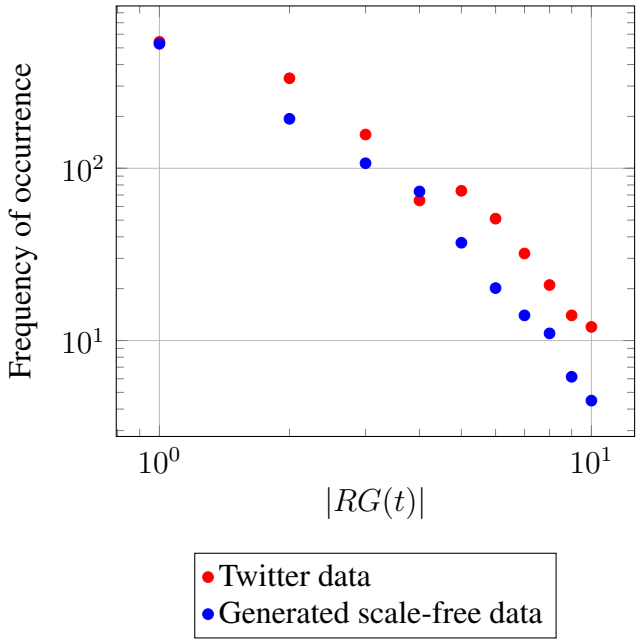


Figure 4.5: Comparison of retweet group size distributions from scale-free graph simulations and data from Twitter's own social graph..

4.1.7 General Comparison of Propagation Characteristics across Different Graph Structures

In this section, three different network structures have been compared, and whilst the path network is very unrealistic as a representation of a social network, the differences in propagation behaviour presented by each do show how the interconnection of users on the graph can have a large effect on the spread of a Tweet. A small set of features to govern retweet features were used in order to accentuate the difference made by the user structures themseleves.

This has demonstrated that, in addition to the processes behind a user's individual retweet decision, the eventual spread of a Tweet also depends somewhat on how the original author's local network is arranged. Thus, the retweet decision of each involved user along with the available information pathways provided by the underlying social structure both contribute to the overall retweetability of a Tweet.

If there are many edges in the network, such as in the case of the random network, then there are many more routes for peopagation to occur down to and from each user, due to the relatively large in- and out-degree of each user node on the graph. This increases the number of users who end up receiving the Tweet and then have the chance to make a retweet decision. This resulted in there being a larger distribution of larger retweet group sizes than smaller ones, before naturally diminishing again. Despite this high throughput of retweets, which provides a high level of information *recall* for the users, the random graph structure is likely to demonstrate a low *precision* in terms of the interestingness of the received Tweets.
This is due to the large number of users having the opportunity to retweet the Tweet, increasing the chance that the 'noisy' information will be filtered through.

The path network demonstrated very poor propagation, and required that its simulation parameters were altered to facilitate retweet behaviour significantly more than in the other graph structures in order to produce any observable pattern. The results showed that propagation down a single allowed chain cannot be an effective way to spread Tweets, as it required each user in the chain to retweet it so that the successive users can have a chance to view it.

Whilst the scale-free network does not have the same general propagation throughput as the random network, it does demonstrate retweet patterns similar to those observed in data from Twitter's own social graph. This complements the findings of [**?**] and [**?**] in terms of online social networks emulating real-life social networks having scale-free properties.
This type of structure supports areas of the graph with denser communities, as is shown to exist by [**?**], and have the potential for facilitating very large numbers of retweets if influential users are involved, but illustrate how Tweets 'travelling' through less dense areas (and less-influential users) will not be as demonstrably popular.

4.2 Using the Social Graph as a Method for Inferring Interestingness

The graph analyses in the previous section have demonstrated a method for generating a $RG(t)$ for a given Tweet, t. Since $t.\text{count}_R = |RG(t)| - 1$, then the same simulation algorithm can be used to estimate a retweet count for a given Tweet. Although it has been discussed that although an individual user's retweet decision does generally imply that user's interest in the Tweet, the overall retweet count of a Tweet is only really capable of denoting that Tweet's popularity. However, this value could be used in tandem with the predicted expected retweet count of a Tweet in order to determine if the Tweet is, in fact, interesting.

This notion is based on the idea that if a Tweet is more popular than expected, then there is something about that Tweet that makes it more *interesting* than similar Tweets that are less popular, such as some breaking news or a link to a controversial article.
For example, consider the case of two Tweets, written by the same author, and both containing the same instances of feature values, such as the inclusion of a URL or a mention. If one of these Tweets achieves significantly more retweets than the other, then there must be some non-trivial feature of the more popular Tweet that makes it stand out to the audience, and thus allows it to be perceived as more *interesting*. This is because the features taken into account are very static, and do address any depth of the actual content of the Tweet.
Similarly, if most Tweets of a user achieve between one and two retweets, then the expected retweet count for this user's future Tweets is likely to be similar. If, however, the author posts a Tweet which achieves an observed total of 10 retweets, then this is more popular than what was expected. If a Tweet achieves one or zero retweets, then this is as expected or less than expected, and is therefore not interesting.

As such, a method is proposed based on the following two criteria;

- observed $t.\text{count}_R >$ expected $t.\text{count}_R \implies t$ is interesting

- observed $t.\text{count}_R \leq$ expected $t.\text{count}_R \implies t$ is non-interesting

Although it was found, in the previous chapter and in other relevant literature, that pseudo-generated scale-free networks can be representative of Twitter's own social structure, a user's actual own local social network would more accurately portray the links between the users surrounding the original author of a Tweet. By constructing a network based on a user's own local network, then the method would effectively be simulating the Tweets' propagation through the edges representing the followships of the actual users in Twitter's social graph.

Thus, in the simulation algorithm, the user in question is u_s, and the intiial value of $S = N^+(u)$. At each timestep, each user in S would have the opportunity to retweet the Tweet, and therefore, by running the simulation, an estimation on the *expected* value for $|RG(t)|$ can be obtained, where $t.\text{author}_O = u_s$.
In particular, the method follows these steps;

1. Select a user, u
2. Collect that user's local follower network
3. Collect a set of that user's recent Tweets
4. Construct a network based on the users and edges of the collected network
5. Simulate the collected Tweets through the constructed network using the simulation algorithm using u as the Tweet's source author user.

This procedure would provide an estimated retweet group size for each Tweet, which could then be compared to the actual observed retweet count of the Tweet on Twitter to help towards deducing the interestingness.

4.2.1 Data Collection

Due to the scaling properties encountered in a breadth-first traversal of Twitter's social graph, it became infeasible to collect a user's local network containing users more than two edge 'hops' away from the source user under the rate limitations of Twitter's REST API.

As previously described, v1 of the REST API allowed 350 calls to the API each hour for each authenticated Twitter account. One call, for example, was required to obtain a list of up to 5,000 user IDs representing the followers of a particular user - the users one hop from the source user. An additional call would then be required to collect each of these user's own followers in order to provide the 2-hop representation of the local network from the source user.

For a user with a follower count of 700, a total of 701 API calls would be required to collect the user's local network within two hops - the one to retrieve the source user's immediate followers, and then one further call for each of the 700 followers. This would take over two hours of collection, and to collect the third hop would require another exponential number of API calls.

If each of the 700 followers of the source user has, on average, 200 followers, then this would require a further $700 \times 200 = 140,000$ API calls, which, in total, equates to over 402 hours of data collection time. Although some follower overlap is likely to be present among the users two hops away, when one considers that this is simply the time taken to collect the local network for *one* user, then it becomes clear that this must still be an impractical approach.

In the previous chapter it was found that the vast majority of retweets do actually occur *within* two hops of the source user, in that retweet groups produced have a maximum path-length of less than three. In addition, as mentioned, online social networks are 'closer' than real-life social networks, and was found to have a value of around four degrees of separation in Facebook. These points help to justify the decision made to classify a user's local network as those users and edges existing within two hops from

the source user.

In June 2012, the Twitter REST API was used in order to conduct a random walk through Twitter's social graph. Starting by selecting an initial user, an edge expressing the followship of a random follower was chosen in order to select the next user. This continued for each of the selected users in turn and, for each user selected, the most recent 300 Tweets and surrounding information was collected along with that user's local follower network within two hops. The friend network (i.e. the outward edges from each user) was ignored, as only the directional outward flow of information from the source user was useful in this experiment.
If, at any stage, the currently selected user did not have any followers, the collection algorithm backtraced to the previous user and another follower was selected instead. The crawler continued until the rate limit for the current request window was met, at which time the current data state was stored, and then waited until the rate-limit was reset before continuing.

The data collection resulted in a set of 33 Twitter users, each with a full local network collected and a set of up to 300 Tweets. In total, around 10,000 Tweets were stored as a result of the crawl to be used in the simulations. It was decided that the previously trained regression model would be re-used as part of the retweet decision engine in this experiment also, and so no further training data was required to be collected. From the Tweets collected, the URL and MENTIONED features could easily be indentified, and the two user features could be extracted under the same process as the one used in the network simulations in the previous section.

For each Tweet collected, a simulation could now be run in order to provide an expected retweet count for that particular Tweet. By comparing this value to the actual popularity expressed by the Tweet, which is returned as part of the standard Twitter API call, an indication of whether or not the Tweet is interesting could be obtained.

4.2.2 Validating the Accuracy of Inference Results

In order to test the validity of the results, it was necessary to use human asessment on each of the evaluated Tweets to check for agreement between the interestingness inferences made by the algorithm and by humans.
Although interestingness is a subjective notion, the validations were carried out in such a way to emphasise a *global* level of interest in terms of the general separation between noisy and un-noisy Tweets.

Crowdsourcing Validations

Crowdsourcing is a technique that has grown in popularity over many domains in recent years, including media, reviews services, sensor networks, and others. Essentially, crowdsourcing involves the use of many people (or, in some cases, devices) providing input or results on a given task.
Services such as Google Maps[2], TripAdvisor[3], and Stack Overflow[4] respectively use crowdsourcing for obtaining information (such as photos) on geographic locations, service reviews, and programming assistance. Its use means that the crowdsourcers can easily receive lots of input with very little additional work, since the load is spread amongst many people.

Crowdsourcing has also proved to be a useful asset in research as it facilitates the harvesting of many inputs, from diverse opinions and views, much more quickly than without it, and it is a useful tool for validating data.
Many crowdsourcing services are active on the Internet to cater for different use-cases. Kickstarter[5] exists as a platform for advertising a product idea or concept with the aim of achieving finance to go towards its production (known as 'crowdfunding'),

[2]http://maps.google.com
[3]http://tripadvisor.co.uk
[4]http://stackoverflow.com
[5]http://kickstarter.com

and Amazon's Mechanical Turk[6] enables crowdsourcers to employ people to carry out tasks.

Mechanical Turk allows crowdsourcers to create small jobs (known as 'microtasks') to be completed by crowdsourcees, known as Mechanical Turk Workers (MTWs), who have an account on the website. The crowdsourcer describes the particular microtask in terms of what is expected of the MTWs and also determines the amount paid for the task. A single microtask completed by a particular MTW is known as a 'judgment', and MTWs are paid for each judgment he/she completes. The crowdsourcer can define certain criteria on the microtasks, such as allowing each MTW to only complete one microtask.

Due, at this time, to Mechanical Turk's availability to only US credit card holders, Crowdflower[7] was used instead to submit the microtasks to Amazon's service in order to be completed by the workers.

Aims of the Validations

The purpose of the use of crowdsourcing was for evaluating the effectiveness of the interestingness inferences made through the aforementioned comparison between an expected and observed popularity of a given Tweet. In particular, of concern was the similarity between those Tweets that the algorithm denoted as interesting and the Tweets that humans found interesting.

Since the accuracy of the various components of the technique could not be known until they were properly validated, it was decided that the crowdsourcing would initially be run as a pilot test in order to identify the presence of any correlations. If this was sufficiently successful, then a further and more rigorous test would take place, involving more crowdsourcees in order to produce a more rigid result.

[6] http://mturk.com

[7] http://crowdflower.com

Constructing the Questions

The microtasks presented to the MTWs each consisted of a question containing five Tweets. The Tweet data from the data collection was compiled into a set of questions in which each of the five Tweets were selected at random.
Each question asked the MTWs assessing the particular question to select which one of the five Tweets was the most interesting and which one was the least interesting, and each of the questions was assessed by at least three different MTWs.

Although Tweet selection was random, Tweets whose content starts with a user's "@" username (i.e. '@-replies') were excluded, since these Tweets typically form part of a conversation between a small number of users and are unlikely to convey any interest to those not directly involved in the conversation. The final validation set consisted of a total of around 4,500 Tweets to be assessed in the questions, and MTWs were encouraged to follow links to any websites or media included in the Tweets' contents as part of their evaluation of each Tweet.

Inference Performance Validation Results

Since at least three separate MTWs were responsible for assessing each question, only Tweets where two or more MTWs agreed on the interestingness of a Tweet were considered, and other instances were discarded. It is assumed that if at least two people agree on a piece of content being interesting, then this provides further strength to the judgment.
Through the retweet simulations and algorithm for each Tweet, an 86% accuracy was achieved in terms of correctly predicting the actual retweet count - the cases where the expected retweet count is equal to the observed retweet count. In around 30% of cases, a Tweet that was determined to be interesting through the methodologies described in this chapter was also verified as interesting by the agreeing MTWs.

This is a relatively low accuracy, and while it does mean that the method was able

to correctly identify an interesting Tweet from a set of five in 30% of cases and the random performance of selecting an interesting Tweet could not reach this accuracy, it is not a strong enough result to describe the method as being suitable in the general inference of interesting Tweets.

As such, further investigation would be required to address the method with the aim of improving this performance.

4.2.3 Improving The Interestingness Inference Performance

The deduction of the interestingness of a Tweet through the comparison of the expected and observed popularity of that Tweet is considered a viable way of addressing the problem for reasons discussed earlier. However, a more convenient and accurate method is clearly required for acquiring the *expected* retweet value.

The issue with the current method is two-fold; as mentioned, lots of data is required in order to reconstruct the Tweets' authors' local networks in which to simulate the Tweets, which leads to the second problem of only being able to simulate Tweets from authors in sparser local networks. Under the current scheme, only users with a small enough local network (i.e. users that have lower folower counts) can realistically be evaluated, due to the collection criteria discussed previously, meaning that the methodology cannot be used in the general case.
Although a high accuracy was achieved in predicting the *correct* retweet counts for the Tweets assessed in this section, most of these Tweets only actually had an observed retweet count of 0 or 1. This is the by-product of the previous issue in that only users with fewer followers could have their Tweets simulated, and these users will therefore typically receive few retweets per Tweet. Ideally, the methodology should have the capabilities to be applied to any type of user and any Tweet on Twitter.

Additionally, this method alone does not make efforts towards evaluating the *level* of Tweet interestingness. Instead of the binary interesting / non-interesting decision, it

would be more useful to award each Tweet a score denoting the estimated interestingness of the Tweet. The further importance and usefulness of this is explained in further detail in the following chapter.

4.3 Chapter Summary

In this chapter, an analysis of propagation through differing structures of user connections on social graphs has been conducted. From this, a potential methodology for inferring interestingness of Tweets has emerged, which, despite being negatively impacted by various factors in its current form, shows promise as a suitable technique towards assisting in this task.

4.3.1 Network Structure Analysis

A logistic regression model was built as part of a simulation algorithm in order to anaylse the propagation characteristics of three different network structures; a path network, a random network, and a scale-free network.
Although the actual retweet counts of simulated Tweets in each network structure are not comparable due to the parameter alterations that were required in order to amplify visible results, the actual *pattern* of propation in terms of the distribution of retweet group sizes was found to be different in each structure and for differing reasons. In addition, the scale-free network was found to express a similar pattern to that observed from the data on retweet group sizes discussed in the previous chapter.

4.3.2 Interestingness Inference Methodology

The model and techniques behind the network structure analyses were then applied to the goal of detecting the interestingness of Tweets based on the comparison of the

expected retweet value, generated through the same algorithm used to simulate Tweets in the network analyses, and the actual observed retweet count of the Tweet.

Validating the methodology showed that the technique is not particularly useful in determining interesting information, and its other drawbacks, such as its application only realistically being available to Tweets from non-influential users, mean that the technique cannot be used in the general case. Further to this, the data collection required is not suitable for quick evaluations and may not remain accurate over time even after collection due to the continuous changing nature of the edges in online social networks as users create and destroy followships. This is particularly impactful in this case as there are many users involved even in a user's 2-hop local network.

In the next chapter, the methodology for generating expected retweet counts is adapted with the aim of improving its validation performance, the ease of preparation through data collection, and of addressing the methodologies current restrictions on the types of users it is suitable for. It is known from work in this chapter that the network structure plays an important role in information propagation, so this and more environmental features are taken on as part of the improvements.
This research then leads to the production of a technique for measuring *how* interesting a particular Tweet is, including the way in which this can be calculated and the different validations performed against it to assess its performance.

Chapter 5

Inferring Interestingness of Tweets based on Information Flow Through the Network

Research in the previous chapter focussed on researching the effect of the social graph on retweet propagation characteristics. From this, a methodology, displaying a range of various shortcomings, emerged based on the models and simulations utilised in the graph analyses. In this chapter, the methodology is modified with the aim of improving its performance and increasing the range of use-cases it is appropriate for. Since the social structure was found to play an important role in propagation, many network and user features are taken into account throughout the improvements.
In addition, modifications are made in order to provide an indication of *how* interesting a piece of information is estimated to be, and more about this particular component is discussed in later sections.

The proposed methodologies also relate to the differences highlighted between a Tweet's raw popularity, as indicated by its retweet count, and hos interesting the Tweet actually is to those who read it. It has been shown that making retweet predictions against models trained with a large number of features can be accurate [**?**], but in this work, the focus is more on the Tweet's content and beyond the static features.
That is, that when comparing Tweet popularity, then there may be some content, either within the Tweet itself or perhaps in a resource indicated by a URL contained in the

Tweet, that makes the Tweet stand out to its recipients and to cause the aforementioned notion of *affective stimulation* [**?**] to its viewers.

Of course, this brings about the notion of information *relevance*, and the fact that the same Tweet could be very boring or irrelevant to one user, and very interesting to another. In this work we focus on *global* (or 'average') interest, where interestingness inferences are made for the general case. It is considered that Tweets that are retweeted more than expected within their authors' local networks, relative to the usual retweet count of the authors' other Tweets, are also likely to be of interest to a wider audience, especially since they are now more likely to penetrate through the social graph enough to be received by users in different communities..

As such, the focus of the work in this chapter is that of adapting the inference methodology in order to develop a technique for accurately *quantifying* the interestingness of tweets. This is concerning universal relevance in terms of highlighting interesting Tweets from the noise. In particular, there are two main improvements of the previous methodology to be made;

- Improve method for generating the *expected* retweet count of a Tweet (in terms of accuracy and range of application)
- Expand the binary retweet interesting inference into a more useful scale in order to support *ranking* of interesting information.

5.1 User Influence

As has previously been posited, of importance to this work is the difference between the retweet count of a Tweet and the interestingness of the Tweet. An example in the Background chapter was provided, which related to the case of Justin Bieber. His account, `@justinbieber`, is one of the most influential on Twitter, with nearly 50

million followers at the time of writing. His Tweets receive an average of around 50-120 thousand retweets per Tweets, and they rarely receive fewer than 40,000 retweets. Since an average Twitter user would generally attract a maximum of a few hundred followers, and would normally receive very few, if any, retweets per Tweet. A particularly interesting Tweet from such a user may be retweeted, for example, between 5-20 times. It is therefore apparent that, in the general case, an uninteresting Tweet from an influential user may receive 50,000 retweets, and an exceptionally interesting Tweet from a less-influential user may be retweeted 30 times. It is therefore clear that user influence dictates that this value cannot alone be indicative of Tweet interest.

However, since interestingness *does* have an effect on an a user's individual retweet decision, this absolute retweet count can be used as part of the method for generating an interestingness *score* for a Tweet.

5.2 Interestingness Scores

To address the notion of interest quantification, a scoring scheme is hereby introduced, allowing certain interesting Tweets to be ranked as 'more interesting' than other interesting Tweets. This, in itself, is an improvement over the previous method, which allowed only for Tweets to be labelled as 'interesting' or 'non-interesting'.

Similar to the previous method's *comparison* between the observed and expected retweet counts, the new scoring technique is based now on the *difference* between the two counts. The general idea and potential use-case for this is that if a score is known for a set of Tweets, then these can be used as a basis for ordering information as part of information retrieval or an information delivery system, where Tweets can be displayed to users in a more useful way and where interesting Tweets could be brought forward to users who don't follow the source user or a retweeter, and thus deliver information to an interested user, yet without him or her having to know about it first.

Essentially, the notion scoring stems from the following scenario. Consider two Tweets, A and B, which have the following properties;

- $A.\text{count}_E = 3000$ and $A.\text{count}_R = 3010$
- $B.\text{count}_E = 5$ and $B.\text{count}_R = 15$

Where $A.\text{count}_E$ represents the expected retweet count of A.

In this case, both Tweets would have been flagged as 'interesting' under the previous scheme (although, in reality, the method would not be able to model users who are typically expected to achieve 3,000 retweets). However, it is clear that, despite the *difference* between the counts being equal, Tweet B's observed retweet count is actually much more significantly proportionately greater than what was expected, and is therefore likely to be more significantly interesting.

Since the proportionate difference is the key to this, the interestingness score, $s(t)$, for Tweet t is hereby simply given by;

$$s(t) = \frac{t.\text{count}_E}{t.\text{count}_R}$$

This provides a positive score where;

$$s(t) \begin{cases} > 1 & \text{indicates } t \text{ is interesting} \\ \leq 1 & \text{indicates } t \text{ is non-interesting} \end{cases}$$

And where $s(A) > s(B)$ implies that A is more interesting than B.

Since this methodology relies on data collection from Twitter in order to obtain the observed retweet counts, it involves extracting a snapshot of the state of the evaluated Tweets at one stage during their lifetime. Since Tweets are not removed over time, unless they are deleted by their author, they can be discovered and retweeted at any time after their composition and posting.

The work in this chapter assumes that the most significant portion of retweet activity for a specific Tweet has already occurred by the time the information on the Tweet has been collected. Indeed, the authors of [**?**] carried out investigative analyses into various temporal retweet brhaviours, and discovered that, on average, that a Tweet receives around 75% of its retweets within the first day of being posted. 50% of the retweets of a Tweet take place within the first *hour* of the Tweet being posted.
Due to this, and to ensure that the retweet count collected is mostly representative of the Tweet's exraploated 'final' retweet count, only Tweets that had been posted at least one day ago were considered for experimentation.

5.3 Further Adaptations of the Inference Methodology

In the previous chapter, it was noted how it was necessary to improve the method used for producing a Tweet, t's, expected retweet count, $t.\text{count}_E$. Problems with the previous method dictated that the method could only work under certain restrictions. In particular, that the user must have a small enough local network (in practice, a follower count of more than 500 or so made the method very unsuitable), and that, due to this, Tweets only attracting very few retweets could effectively be simulated. In addition, the interestingness inferences made were not significantly accurate, although this is likely due to a combination of the aforementioned issue providing much less room for error and the fact that the interestingness decision was only binary.

A new method is hereby proposed for carrying out the prediction for the value of $t.\text{count}_E$. This method is immediately more superior to the previous, as only a very small amount of data (if any) is required to be collected from Twitter. This means that inferences on Tweet interestingness could be made on demand[1].
Essentially, the method involves creating a classifier model capable of producing a base-line expected retweet count for a given Tweet and its relationship with its author.

[1]Not 'live' due to retweet action relies on time to occur.

In this case, the classifier would be trained with the Tweet's actual retweet *count* instead of the binary retweet decision used previously, and it would not require the simulations of the user's local network. Many more features regarding the Tweet, and its conetent, and its author are used to represent the particular user-Tweet information required for generating the predictions.

Since the graph structure clearly has an impact on message propagation, then it was felt that a significant consideration should be made towards including features relating to the interconnection of users, such as follower counts, Tweet rate, and information on a sample of friends and followers. More detail on the features used is provided in later sections.

In general, the newly proposed methodology follows these basic steps:

1. Collect sufficent data from Twitter to train a classifier with an appropriate set of features. The trained model is known as the 'global' model;
2. Obtain a Tweet, t, and extract its own features as well as information about its author and its author's network properties;
3. Classify the Tweet's features against the trained classifier to produce a retweet count prediction, $t.\text{count}_E$ for this feature instance;
4. Calculate s(t) through using this $t.\text{count}_E$ value and the known $t.\text{count}_R$.

In addition to this 'global' model, a 'user' model was proposed to be built for each user being evaluated. This user model would be much smaller, as it would only contain information on that user's historical Tweets, but would be capable of providing a second value for $t.\text{count}_E$. With two such values, two scores could be generated by comparing the static $t.\text{count}_R$ to each in turn.

As such, the two scoring mechanisms work as follows;

$$s_G(t) = \frac{t.\text{count}_R}{t.\text{count}_{E1}}$$

$$s_U(t) = \frac{t.\text{count}_R}{t.\text{count}_{E2}}$$

5.4 Retweet Volumes as Nominal Attributes

Most machine learning classifiers are not useful in accurately predicting the outcome of a feature of a large-ranging and continuous data type. Instead, the performance can be greatly improved when predicting from a limited range of discrete ranges, or 'nominal' data.

Thus, in order to help improve the accuracy of $t.\text{count}_E$ predictions, it was decided to convert the retweet count feature into a nominal data type for the purposes of training the model and making classifications. By 'binning' the retweet counts into categories representing interval ranges, there would be fewer outcome possibilities, and thus the *confidence* of classifcation could be greater.

The values for $s(t)$ would then be determined through the ratio of $t.\text{count}_R$ to the upper-bound of the nominal range category containing $t.\text{count}_E$.

5.4.1 Binning the Retweet Counts

Since a trained classifier is only generally able to make predictions on features and values it has prior knowledge of, the bin ranges for each category must be equal in both the training feature data and the testing feature data. If the available nominal values for an instance feature representing a Tweet has a different set of category ranges to that in the trained classifier model, then it is likely that a prediction cannot be generated for this instance. It was therefore necessary to consider this when determining a method for binning the retweet counts.

There are various ways in which the counts could be binned, and all begin with a

decision on the number of bins to use. The varying performance of this factor is considered later.

Initially, retweets were binned in a *linear* fashion. That is, that the full range of retweet counts in the training set was calculated and then split into bins such that each category had an equal a range as possible. If there were no cases where $t.\text{count}_R = 0$, then a category representing $[0, l)$, where l is the minimum value for the lowest range, was pre-pended to the set of available nominal categories. Similarly, in all cases, the interval $[m + 1, \infty)$ was appended to the set of categories, where m is the maximum value in the highest range. This dictates that no Tweet in the training set can have a value for $t.\text{count}_R$ in this category, and thus this allows any Tweet to potentially have $s(t) > 1$. For example, if a training set of Tweets had a total range of values for $t.\text{count}_R$ being between 1 and 20 was binned into four ranges, then the following interval categories would be applicable:

$$[0, 1)[1, 6)[6, 11)[11, 16)[16, 21)[21, \infty)$$

Since the distribution of retweet counts (expressed through retweet group sizes) is known [**?**], then it is clear that this binning methodology would produce bins containing a very non-uniform distribution of Tweets, where the lower bin ranges would contain many Tweets and the cardinality of each category would decrease exponentially as the ranges become higher. This means that there would be fewer feature instances representing Tweets with larger retweet counts.
Indeed, when training classifiers and running cross-validations on these, this binning scheme demonstrated a high accuracy of predictions on Tweets with lower values for $t.\text{count}_R$ and a low accuracy for Tweets with higher counts. It would be more appropriate, and better address the desire for more universal use-cases expressed earlier in this and the previous chapter, if the accuracy of predictions could be more uniform across the bin ranges.

Various other methodologies were implemented, which eventually evolved into a histogram-based responsive binning algorithm. Essentially, this algorithm involved is based around

the initial calculation for the projected size of each bin, which is based on the total number of Tweets to be categorised and the target number of bins. Each bin was then filled according to the interval range specifying the bounds of that bin, and in such a way such that each retweet count frequency would only be present in one bin. For example, all of the retweets achieving one retweet would be placed in the single bin encompassing this value.

As such, after the intervals representing the bin bounds have been produced, then these represent the nominal categories for the retweet count feature in each instance for training and testing against the classifier.

Algorithm 2 Algorithm for producing intervals for bin categories for $t.\text{count}_R$ values.

procedure GENERATE_INTERVALS(set of Tweets T, number of bins B)
    $C \leftarrow$ empty list ▷ To hold ordered retweet counts
    $I \leftarrow$ empty list ▷ To hold bin range intervals
    for all $t \in T$ **do**
        Add $t.\text{count}_R$ to C
    end for
    Sort C into ascending order
    $M \leftarrow \max(C)$ ▷ Highest instance of $t.\text{count}_R$
    $T\text{Sum} \leftarrow \frac{|C|}{B}$ ▷ Number of Tweets in each bin
    $H \leftarrow$ empty dictionary ▷ Histogram of retweet count distribution

    for all $c \in C$ **do**
        if $c \in H$ **then**
            Increment H_c
        else
            $H_c \leftarrow 0$
        end if
    end for
    for all i in range $M + 1$ **do**
        if $i \in H$ **then**
            $s = s + H_i$
        end if
        if $s \geq T\text{Sum}$ **then**
            Add i to I
        end if
    end for
    Return I
end procedure

This new responsive method more readily supports more uniform bin sizes, and copes with this by expressing exponentially larger bin *ranges*. As such, the distribution of bin sizes is generally described by a distribution simular to that shown in Figure 5.1. As with the linear method, the interval $[0, l)$ is pre-pended, where necessary, and $[m + 1, \infty)$ is always appended in addition to the intervals produced by the algorithm.

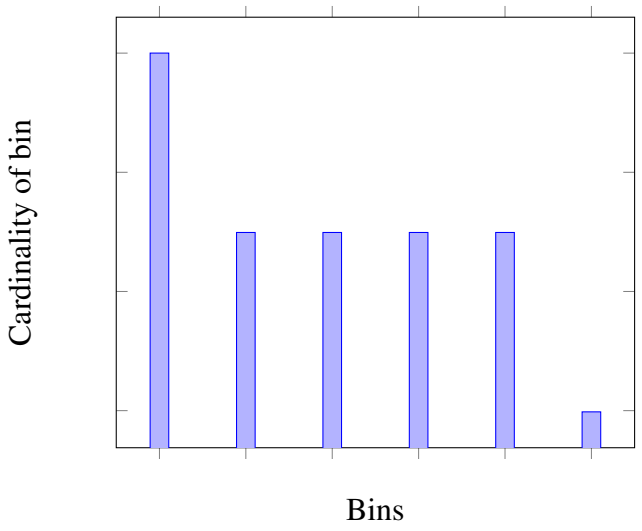


Figure 5.1: Example distribution of retweet count bin cardinalities under the responsive binning algorithm.

This method is responsive in that the bin ranges adapt to the variety and number of retweet counts available, and always attempts to produce a similar number of bins to what is requested. However, due to the disproportinately large number of small retweet groups, the bin sizes cannot be entirely uniform and means that the number of intervals returned will be smaller than the number requested.

This also stems from the fact that a single retweet count cannot exist in more than one bin concurrently; for example, if the interval $[0, 2)$ existed in a scenario, and the number of Tweets with retweet count equal to 0 or 1 is greater than the value for TSum, which is often the case, then this will result in a larger bin. Without this particular feature, a Tweet may be categorised into more than one bin, causing the prediction accuracy to be reduced.

Due to this dynamicity, the bin ranges and cardinalities produced by the algorithm vary across different datasets. As a result, the nominal bin categories generated for producing the value for $s_U(t)$ from the user model trained from the complete set of collected Tweets posted by $t.\text{author}_O$ would be different from those categories generated for a different user. The intervals in each bin category are therefore reflective of the various different number of retweets that each author's Tweets are likely to receive.

5.5 'Twitter is a Memepool'

In 1976, Richard Dawkins coined the term 'meme' to be defined as a "unit of cultural transmission" [**?**]. The general idea behind memetics is as an analogy to biological genetics except, unlike genes, memes are entirely non-physical and represent a cultural idea or aspect or another human-based behaviour. The rise of social networks on the Internet has allowed the spread of memes to grow to the extent that they are sometimes now even *represented* by physical constructs, such as images.

In genetics, a gene is a physical entity containing information and instructions. It is a unit of genetic inheritance, in that they are passed from parent to offspring through the act of reproduction, and the result of an organism having a gene will express the features represented by that particular gene. These genes contain instructions that make up the features of an individual, such as physical characteristics like eye colour and height, and non-physical characteristics, including various aspects of personality.
Organisms exist in an evironment that also has features, such as humidity, altitude, temperature, relations to other organisms, and so on. If the genes of an organism are such that they cause the individual to be well-suited to its environment, then that organism has a better chance of survival and, therefore, a better chance of achieving reproduction.

Memes are similar in that they are effectively made up of a set of features, or 'memome', such as the wordings of a particular phrase, or their relevance to other cultural

aspects. These enable the meme to be less or more likely to be replicated in different environments, which is made up of the humans exposed to it and the interactions between them. For example, an Internet meme relating to the Star Wars movies would likely have a greater chance of being reproduction, through discussion and reposting, in an environment comprising a set of science-fiction fans than when amongst more mixed-interest groups.

The meme is also a useful analogy in this thesis when describing the way in which Tweets undergo replication within Twitter. Like a meme, a Tweet has a specific set of features, such as the text it contains, the inclusion of any mentions or a URL, and so on, and it exists within an environment consisting of a set of interconnected users on the Twitter social graph.
A particular Tweet would generally have a greater chance of 'surviving' and being replicated, through the act of retweeting, amongst certain users intereconnected in a particular way than in other environments.

As such, the Tweet features are analogous to the *genes* of a genome, and the arrangement and type of users on the social graph that receive the Tweet and have an opportunity to assist in its propagation comprise the Tweet's *environment*. Since the environment has previously been found to have a large effect on propagation, then these features are useful aspects to include as part of the improved methodology covered in this chapter.

5.6 Generating Values for $t.\mathbf{count}_E$

In order to generate the estimated retweet counts, a trained machine learning classifier is needed to make predictions on a set of feature instances. This section covers an overview of the classifier used for this purpose including a justification in terms of an analysis of its performance.

5.6.1 Machine Learning Classifier

An overview of machine learning classifiers and their processes was provided in the previous chapter. In that case, a logistic regression was used to generate a prediction on a binary retweet decision based on a small number of features. If the retweet count for the Tweet being trained or tested was greater than zero, then the retweet decision would be positive (TRUE). Otherwise, the decision was negative (FALSE).

Presently, the new methodology involves the prediction of a retweet count category from a set of nominal values of greater cardinality than two. As mentioned, the instances of a particular Tweet and its evironment are categorised based on the value of the retweet count of the Tweet. Although this means that a degree of accuracy is sacrificed when training the classifier, it does mean that there are fewer categories for predictions on test Tweet feature instances, providing a higher confidence in each prediction made.

The Bayesian network machine learning classifier was elected for use for the purposes required in this chapter. Use of this classifier in the social media domain is more rare than other classifiers, such as those involving a regression or a decision tree, but was selected for the various reasons highlighted later in this section.

The Bayesian network is an unsupervised classifier since its learning algorithms do not simply determine the class of the outcome, the retweet count, from the attribute features alone [**?**]. Instead, a probabilistic graph is constructed based on the dependencies between the variables. The variable attributes form the nodes of the graph and edges between the nodes denote the dependencies (or lack thereof) between them.

Thus, in the case of this research, the various Tweet and envirnomental features, including the nominal retweet count, form the nodes in the Bayesian Network. When forming the graph through training, the dependencies and their probabilistic weightings are adjusted so that an expected value for the retweet count can then be 'predicted' from the values of all the other variable attributes.

5.6.2 Classification Performance

When selecting classifiers, the Weka[2] machine learning and data mining toolkit was used to evaluate the relative performance of various types of appropriate classifiers for the task. The classifiers were selected to cover a sample of the range of available classifier categories. Whilst some types may work inefficiently in this scenario, it is likely that they are more efficient when employed in different use-cases.

Although the accuracy of prediction was important, it would also be useful for the classifier to be *efficient* in training its model and when testing future instances against it. This is so that this method could be used to produce interestingness inferences on demand and to further improve on the methodologies used in the previous chapter.

Classifier	Precision	Accuracy (recall)	training time (secs.)
Simple logistic	52%	56%	528
Logistic	62%	56%	18
SMO	51%	55%	1384
Naïve Bayesian	50%	44%	0.13
Bayesian network	62%	64%	0.54

Table 5.1: The training performance of different machine learning classifiers.

The Bayesian network was found to be accurate and time efficient when evaluating the performance of the set of classifiers. A dataset, obtained as part of the general data collection (please see the relevant section below) and which contains a set of over 57,000 Tweets, was identically used in each of the analyses. For each Tweet instance the same retweet count binning scheme was used, and each classifier performed the same number of cross-validations against the same dataset in order to obtain the precision and recall values.

Although the dataset used in this analysis is not the complete set used in practice, the cardinality of the dataset was sufficient to cause the outputs to be indicative of the

[2]http://www.cs.waikato.ac.nz/ml/weka

Bayesian network's relative advantages over the other assessed classifiers.

5.6.3 Varying the Cardinality of Nominally-Categorised Retweet Counts

Applying the continuous retweet count values to produce a set of nominal categories representing interval ranges of the retweet counts requires a certain balance.
By reducing the number of target category bins then the classification accuracy increases, but the level of applicability of the eventual interestingness score for the wide range of retweet counts observed would be reduced. Inversely, with too many bins, the classification accuracy reduces, as there would be fewer instances in each category, yet the scores would be apliccable to a wider range of retweet counts.

Target category count	Resultant category count	Precision	Accuracy (recall)
1	1	100%	100%
2	2	89.3%	89.3%
5	4	78.8%	74.5%
10	7	68.6%	65.7%
15	10	61.2%	56.4%
20	12	59.1%	52.9%
25	15	51.4%	47.5%
30	18	49.3%	45.3%
35	21	47.2%	43.2%
40	23	46.2%	42.5%

Table 5.2: The effect of varying the number of nominal categories representing retweet counts on the classification performance using a Bayesian network classifier..

Clearly, by increasing the number of nominal categories used, then the relative number of feature instances in each eventual interval decreases. These bins represent the nominal categories that each feature instance is classified as in relation to the predicted

retweet count of the instance. Table 5.2 outlines the decrease in classification accuracy observed with increases in target bin count. The dataset used consisted of nearly 67,000 Tweets, also as part of the ongoing general data collection discussed in later sections, and the classifier used to conduct the analysis was a Bayesian network through cross validation.

It has already been discussed that the resultant bin count is usually likely to be less than that requested of the binning algorithm. This is due to the long-tail distribution of retweet counts referred to previously.
In the coming experimentations the algorithm tried to produce, where possible, about 10 nominal ranges for use with training and testing against the general global dataset for the purposes of generating the global expected retweet count. Since each user's own retweet count ranges were different, the number of categories were calculated individually for producing the user-centric expected retweet counts as part of calculating values for $s_U(t)$.

5.7 Training and Testing Against the Bayesian Network Classifier

This section discusses the processes used behind the calculation of interestingness scores for Tweets through the generation of expected retweet counts using the methodologies outlined in the previous sections. Particular focus is lended to the data collection and the features extracted through the resultant data corpora.

5.7.1 Collecting the Training and Testing Data

In order to train the model on a set of Tweets and then use it to make predictions, data was required for collection from Twitter. This data could then be divided up when required, as described below.

Since, in this case, it was necessary to collect the Tweet data along with each Tweet's numeric retweet count, rather than the binary nominal yes/no required in the previous chapter, only the retweets of a particular Tweet that had been created using the button method could be considered. This is because a Tweet's retweets executed using the manual copy and paste method do not contribute to the Tweet's official, and observable, retweet count that is returned from Twitter's API. This is not considered to be a limitation, however, since this factor is used consistently through the training and later evaluation of the trained model.

In March 2013, a random walk was conducted through Twitter's social graph using v1.1 of Twitter's REST API. Although this date was before the mandatory transfer to this version of the API, the crawler method was used in preference over collecting from the public API, which was deprecated and removed in v1.1, so that user data could be collected, as described, for the environmental training features.

The walk originated with one Twitter user, and each stage consisted of focusing on one user before selecting another one from the followers of the currently focused user. As such, the crawler is very similar to that used in the latter sections of the previous chapter.

At each step of the crawl, a set of the most recent Tweets authored by the current user were collected. The number of Tweets obtained for each user had various dependencies, such as the user's Tweet-posting frequency and the number of Tweets in total posted by the user. Usually, several hundredf Tweets from each user were yielded. In addition to the Tweet data, information on the user itself was collected as well as on a sample subset of up to 100, if they exist, of its friends and followers.

A sample subset of friends and followers was used instead of the complete set for the purposes of efficiency and to address the associated limitation in the previous interestingness inference methodology, yet it still provides an example snapshot of up to an additional 200 users in the author user's local network in order to provide some idea of the activity within the local network both upstream and downstream from the au-

thor user. Around ten API calls were required to obtain this information for each user, giving it immediate advantages over the older method.

The data collection crawl resulted in a dataset containing around 241,000 Tweets authored by 370 unique Twitter users. Of those Tweets, around 90,000 were cases in which the retweet count was greater than zero. The partial datasets as subsets of the complete set, obtained up until various intermediate points of the entire collection, were those used by the classifier and binning performance analyses in the previous section.

Importantly, Tweets from many different types of user were collected; from less-active users with very few followers and friends to influential users and celebrities with millions of followers and achieving many thousand retweets. The collection of this range of users will help demonstrate if this new methodology is able to correctly assess a wider range of users and Tweets.

5.7.2 Data Corpora

After collection, the complete data was divided into two datasets; a training set, consisting of 90% of the entire set, and a testing dataset, consisting of the remaining 10%. The original set was divided in such a way as to ensure that all of the Tweets authored by one particular user existed in only one of the two resultant datasets. After being used to train the Bayesian network model, the larger dataset was then discarded from use for the rest of the experimentation.

As has been previously mentioned, of interest is the generation of *two* interestingness scores for each Tweet, t; one based on a comparison between $t.\text{count}_R$ and an $t.\text{count}_E$ produced from the global model, and the other between $t.\text{count}_R$ and an $t.\text{count}_E$ produced from $t.\text{author}_O$'s user model. As such, each Tweet requires the two models in order to provide the predicted values for $t.\text{count}_E$.

To assist with this, individual user datasets were extracted from the testing dataset,

each containing information only on that particular user and its local network and its Tweets, from which individual user Bayesian network models could be trained.

The complete testing dataset is referred to as the 'global' corpus of Tweets, and each individual user dataset is known as a 'user' corpus.

5.7.3 Features

Producing the instances used for testing and training the Bayesian network models involved the extraction of various features from the global and user datasets. Generally, each feature falls into one of three categories; the network features ('environment'), the Tweet features ('genome'), and the author features (representing the author of the current Tweet). The nominalised retweet count is categorised as a Tweet feature.

Generally, the Tweet features follow the same notions as those used in the previous chapter in that they are static and generally binary features describing various aspects of the Tweet's content and metadata. The network features are more variable and describe the ways in which the author's local network is constructed and the activity within it.

Each Tweet is represented by an instance of a complete set of features relating to that Tweet, its author, and its author's local network. As a result, feature instances representing Tweets authored by the same user will share the same values for their network and author features.

Features for the global corpus model

The global corpus model is the Bayesian network model representing the classifier trained from the complete training dataset. In this case, a total of 31 features, outlined in Table 5.3, were used to train the classifier.

As such, there were around 217,000 Tweet instances using this feature scheme used for traning the global classifier.

Feature category	Feature	Feature data type
Tweet ('genome')	mention	{True, False}
	Tweet length	real (numeric)
	url	{True, False}
	hashtag	{True, False}
	positive emoticon	{True, False}
	negative emoticon	{True, False}
	exclamation mark	{True, False}
	question mark	{True, False}
	starts with 'RT'	{True, False}
	is an @-reply	{True, False}
	retweet count	**[dynamic nominal]**
Author	follower count	real (numeric)
	friend count	real (numeric)
	verified account	{True, False}
	status count	real (numeric)
	listed count	real (numeric)
Network ('environment')	max. follower count	real (numeric)
	min. follower count	real (numeric)
	avg. follower count	real (numeric)
	max. friend count	real (numeric)
	min. friend count	real (numeric)
	avg. friend count	real (numeric)
	avg. status count	real (numeric)
	proportion verified	real (numeric)

Table 5.3: Features used to train the model from the global data corpus.

The network features listed apply to both samples of the followers and friends retrieved for each author user during the data collection. For example, the first feature of this category, 'max. follower count', represents two features referring to the maximum follower count observed across the sample of the user's followers and the sample of

the user's friends respectively.

It should be noted that although the Tweet features, aside from the retweet count as has already been discussed, are permanent after the Tweet has been created and posted, the author and network features are more dynamic due to the continuous mutations in the social graph as edges representing followships are constantly being formed and broken between the user nodes. In this thesis, it is assumed that changes to the features representing these factors were not significant over the period of posted Tweets for each user, and the effect is minimised through consideration only of the recent Tweets of each author user.

Features for individual user models

Since the author and network features have identical values in the instances representing all of the Tweets from one particular user, then these features were not considered when training and testing using the user models.
As such, the 10 Tweet features were those used in the feature instances in training, and testing against, each user model.

5.7.4 Testing Against the Trained Classifier

Once the feature extraction was completed and the instances were built for the global training dataset and each individual user set, the models were trained as described above.

In order to produce the expected *global* retweet counts, each Tweet $t \in T$, where T represents the entire testing dataset, had its features extracted, less the retweet count nominal category, and was evaluated against the global model. This classified each Tweet into one of the categories given by that Tweet's predicted retweet count, and the top interval of the expected retweet outcome category became the *expected* retweet count for that particular Tweet.

Similarly, the expected *user* retweet counts were produced in the same way, but instead each Tweet was classified by the user model associated with that Tweet's author.

In each case, the two interestingness scores for each Tweet could be calculated, based on the process described earier, using the ratio between the Tweet's two expected retweet counts and its *observed* retweet count stored as part of the data collection from Twitter. This meant that each Tweet $t \in T$ had two numeric scores, $s_U(t)$ and $s_G(t)$, assigned to it.

5.8 Initial Validations of the Scoring Methodologies

Similar to in the previous chapter, tests are required in order to ensure the validity of the interestingness scores applied to each of the Tweets in the testing dataset. By running these validations, the relative performance of the scoring mechanism can be assessed, and the comparative performance of the two scores, $s_U(t)$ and $s_G(t)$ can be evaluated.

5.8.1 Planning the Validations

It was decided that crowdsourcing through Crowdflower and Mechanical Turk, again, would be used to validate the new scoring mechanism, as this would facilitate interestingness evaluations from a wider range of human input. The MTWs taking part would not be associated with the collected Tweets in any way, and thus this assists in the identification of the non-noisy Tweets that are 'globally' interesting and are those that the scores have theoretically determined as 'interesting'.

Certain Tweets and users were removed, at this stage, from the dataset of Tweets to be assessd by the MTWs. Since the Tweet data was collected through a random crawl through the Twitter and no checks were placed on the crawler at that stage, there was no governance over the content of the text in each Tweet of the data. Thefore, users who frequently used offensive phrases or wrote Tweets in non-English had their Tweets

removed. The reasoning behind the latter is based on the fact that the Mechanical Turk microtasks were submitted to be completed by people living in the USA.
As before, individual Tweets that were '@-replies' were also stripped so that only Tweets intended to be broadcasts were included in the final MTW test set.

5.8.2 Validating the Methodology Outcomes

In the context of this validation scheme, the MTWs were random account-holders on Mechanical Turk and had no connection to the Tweets they were evaluating. By not determining the humans to make the assessments, a more diverse opinion on the interestingness can be achieved, as the different users will have varying considerations on what contstitutes 'noise' and will therefore form a more diverse opinion and further reinforce a decision when multiple MTWs form agreements on what is interesting.

The validations were carried out such that the MTWs were presented with a series of questions, each of which consisting of five different Tweets from *one* specific author. As such, Tweets were assessed against others than had been posted by the same user. In each question, the MTWs were asked to select the Tweets that they consider to be the most interesting of the group, and that they must select at least one Tweet for each question. For each judgment, where a judgment is one question answered with one or more Tweets selected, MTWs were paid $0.05.

The test was conducted under the condtitions of a randomised controlled trial. To this end, each Tweet was assessed in three different contexts, in that it would appear in three different questions alongisde four other randomly chosen Tweets, and that each question would then be judged by three different MTWs.

From the stripped testing dataset, 750 Tweets were selected, at random, to be filtered by the author user into the questions to be assessed on Mechanical Turk. Since each Tweet was to appear in three different questions and since each question consisted of five unique Tweets, then this resulted in a total of 450 different questions. Each of

these questions was then judged by three different MTWs.

5.8.3 Outcomes From the Validations

The validation test involved contributions from 91 different MTWs, demonstrating the wide diversity of human input attainable validations employing crowdsourcing in this way. From these MTWs, 325 of the 450 questions in total asked had responses where a Tweet was selected with a confidence of two-thirds or greater. Since the MTWs had the opportunity to select more than one Tweet of each question to be the most interesting, there were 349 Tweets of the original 750 Tweets, denoted as $T' : T' \subset T$, that were selected as sufficiently interesting by the MTWs. Tweets selected from individual questions that did not have sufficient confidence were discarded.

The remainder of this section analyses the validation data in various ways to demonstrate the strengths and weakenesses of the interestingness score inferences.
Of immediate notice was the comparitive difference between the two different scoring mechanisms for each Tweet t; $s_G(t)$ and $s_U(t)$. The inference validation results are not significant between the use of the two scores in any of the analyses conducted. As such, the following analyses concern only the use of $s_G(t) \forall t \in T'$.

General Performance

Of the subset T', the scoring mechanism found 140 of the Tweets to have a value of $s_G(t) > 1$, and thus inferred as interesting. Of these, 65% were agreed on as interesting by the MTWs. The performance of the $s_U(t)$ was worse in providing a 55% agreement, resulting in a general of 60% agreement on the mean of the two scoring schemes.
It is also demonstrable that the proportionate frequency of Tweets with higher values of $s_G(t)$ is greater in the subset T' than in T. This implies that, on average, the MTWs were selecting and agreeing on Tweets being interesting that had a higher score than those that were *not* selected. Further to this, there is a greater proportion of Tweets

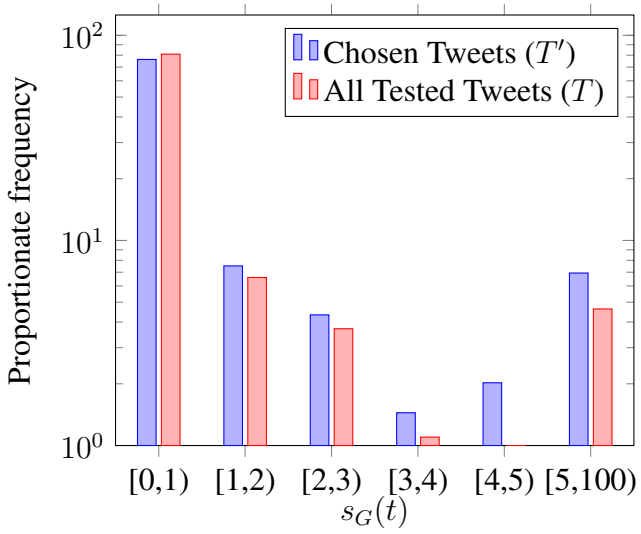


Figure 5.2: Proportionate frequency distribution of $s_G(t) \forall t \in T$ compared to only those $s_G(t) \forall t \in T'$.

with lower scores ($0 \leq s_G(t) < 1$) in T than in T', and a greater proportion of higher values for $s_G(t)$ in T' than in T.

This means that, in general, the humans were marking a greater number of Tweets as interesting that were inferred as interesting by the scores than ones that *weren't* inferred as interesting. Although this demonstrates a clear advantage on the binary inference of interestingness over the methodologies in the previous chapter, this analysis does not consider how well the scheme is able to *rank* Tweets in order of interestingness.

Per-Question Performance

In order to assess the ability of the scores to effectively rank Tweets in order of inferred interest *level*, the Tweets were studied on a per-question basis.

These questions are those that were assessed by the MTWs and where a particular question, $q \in Q$, where Q is the set of all 450 questions, is comprised of a set of Tweets such that $|q| = 5 \forall q \in Q$.

In order to conduct this analysis, each question q had its five Tweets $t \in q$ ranked in

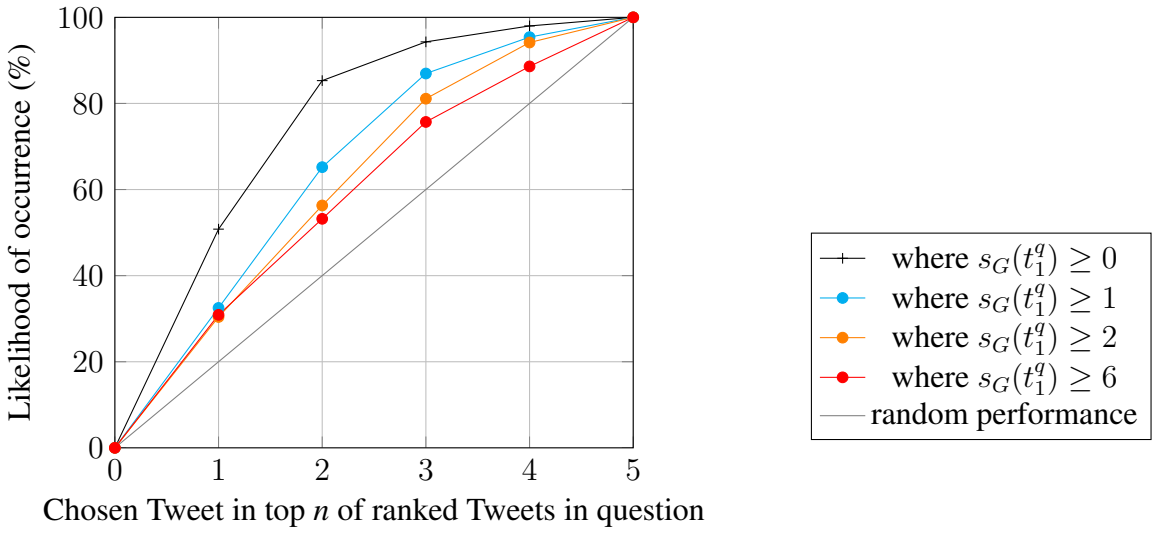


Figure 5.3: The probability of a selected Tweet's $s(t)$ being in the top *n* of *ranked* Tweets for that question. Also illustrating the effect of raising the minimum allowed $s_G(t_1^q) \forall q \in Q$.

order of ascending $s_G(t)$, such that;

$$q = (t_1^q, t_2^q, t_3^q, t_4^q, t_5^q)$$

and where;

$$s_G(t_1^q) \geq s_G(t_2^q) \geq ... \geq s_G(t_5^q)$$

In cases where $\sum_{i=1}^{5} s_G(t_i^q) = 0$, then that q was removed from Q and discarded for the analysis.

Calculations were then made on the number of times the MTWs selected a Tweet as interesting that appeared in the top n of the ranked list Tweets for each $q \in Q$. For example, in the case of $n = 2$ with a set of ten questions, if the MTWs selected one of the top two of Tweets in five of ten cases, then the chance of this occurring is 0.5. Figure 5.3 illustrates the relationship between increasing values of n and this calculation on likelihood of selection. Although the 'random' performance represents the relative likelihoods of a random selection being made when only one Tweet is selected from each question, the vast majority of questions were actually answered with only one

Tweet selected. Further analysis to cover the consideration of this particular point is conducted later on in this chapter.
Further to this, the minimum allowed value of $s_G(t_1^q)$, which represents the highest score of all $t \in q$, was varied with the aim of demonstrating that the detection of more interesting Tweets can be more accurate when the relative score *range* of a particular question is more disparate.

When considering cases where the most interesting Tweet in a particular question is, indeed, inferred as interesting ($s_G(t) \geq 1$), then the MTWs selected one of the *top two* Tweets in around 66% of cases, and they selected one of the *top three* ranked Tweets in 87% of the questions. This demonstrates the method's ability of being able to effective rank Tweets in agreement with humans in identifying information from the noise around it.

Probability of Selection

A further analysis in this section is with regard to the *probability* of a particular inferred-as-interesting Tweet being selected as interesting by the MTWs. It is demonstrable that that, in general, the chance of a particular MTW deciding that a Tweet, t, is interesting becomes greater as the value of $s_G(t)$ increases.

Although the cases of Tweets where $s_G(t) > 4$ are excluded, for the purposes of noise reduction from fewer samples, Figure 5.4 shows an observable increase in probability of selection as the score increases. This pattern is particularly applicable in the score interval of 0-1, which represents the range of Tweets that the scoring scheme has inferred as uninteresting to those that achieved a correctly-predicted popularity, and are thus 'as expected' in terms of interestingness.
The analysis is also clear that Tweets with an inferred interestingness score of three or more are not significantly different from one another in terms of the level of interestingness assigned from the 'real' human judgment.

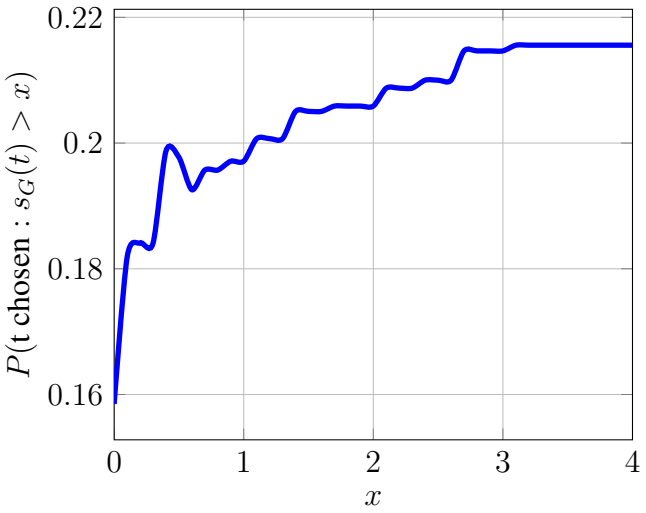


Figure 5.4: Cumulative frequency representing the probability that Tweet t is chosen provided that $s_G(t)$ is greater than a given value, x.

Discerning Interesting Information from Noise

The metrics behind the human selection in determining interesting Tweets is the final analysis conducted in this section. Of particular concern is the varying likelihood of *agreement* between the MTWs and the relative properties of the Tweets and their scores in each question in different decision scenarios.

The notion of score *disparity* is used to determine the difference in interest between a set of Tweets presenting with a range of different interestingness scores. To this end, each question asked of the MTWs has a disparity associated with it. The absolute score disparity for a question, $q \in Q$, is calculated such that;

$$d_G(q) = \max(s_G(t)) - \min(s_G(t)) \forall t \in q$$

A confident answer to a question is one where the particular question had at least two of its three assessing MTWs select the same Tweet as interesting. Since an MTW could select more than one Tweet from each question, then each question may, in fact, have more than one confident answer. Table 5.4 illustrates how questions with varying

Num. confident answers in q	min. $d_G(q)$	max. $d_G(q)$	avg. $d_G(q)$
0	0	846	**17.6**
> 0	0	1445	32.1
1	0	1445	**34.3**
> 1	0	4	0.647
> 2	0	0.55	0.204

Table 5.4: Illustrating trends between the absolute $d_G(q)$ with the varying number of confident answers made in q. Entries in bold are used to highlight interesting values.

score disparities can have an effect on the probability of MTWs being able to make a confident decision.

The results show that the average $d_G(q)$ of all $q \in Q$ is roughly double in cases where a question is answered with precisely one confident Tweet than in cases where there was no confident answer made at all. This indicates that a wider range of interestiness of information is helpful to humans in identifying the content they'd prefer to read. If several pieces of information were displayed to a user that had more similar scores, and are therefore more equally interesting (or uninteresting), then it becomes more difficult for an agreement to be made between the different users on which particular piect of information is the *most* interesting.

Further to this, the average disparity is much lower in cases where the question had multiple confident answers made. In these cases, the MTWs have selected more Tweets as interesting, which helps to reinforce the above point in that identifying one particular piece of information as the most interesting is more difficult when the pieces are all of similar interest levels. Indeed, in questions where this is the case, MTWs have selected, and agreed on, multiple Tweets.

Table 5.5 highlights the effect of disparity on human selection also through demonstrating that in *all* $q \in Q$, and not just those that have been confidently answered, the score disparity between all of the Tweets in a particular question that were selected,

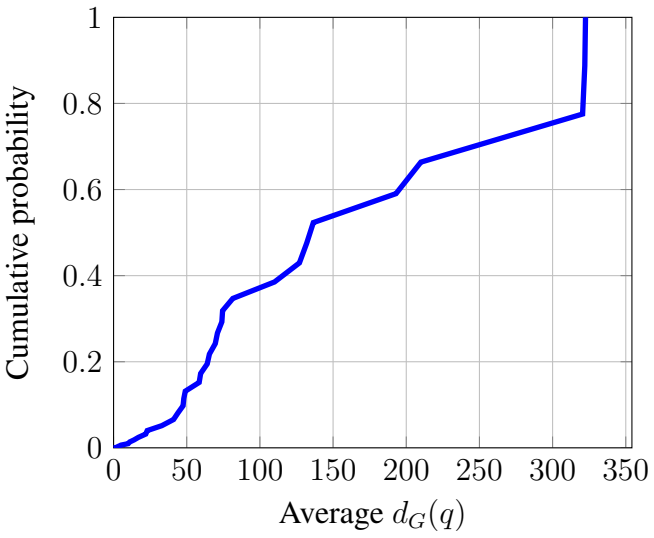


Figure 5.5: Cumulative probability of a confident selection being made for question q with varying $d_G(q)$.

$d(q)$, is much smaller than the general disparity of the entire question, $d(q)$. In this case, the disparities are dependent on the particular scoring scheme declared in order to highlight the differences between the two scores, and where;

$$s_{\text{avg}}(t) = \frac{s_G(t) + s_U(t)}{2}$$

This feature is particularly observable in cases where a question consists of a few Tweets having similarly high scores amongst Tweets with collecively lower scores. Therefore, inferring the interesting Tweets is easier, demonstrated by the scores of selected Tweets being generally higher, but discerning one *most* interesting Tweet is not as trivial. For example, the results show that, on average, the disparity of the global scores across selected Tweets is around 57% of the disparity across *all* of the Tweets in a particular question.

	$s_G(t)$	$s_U(t)$	$s_{\text{avg}}(t)$
$d(q)$	62.4	4.7	33.3
$d_{\text{sel}}(q)$	35.3	3.1	19.0
Ratio	57%	66%	58%

Table 5.5: Comparing the disparity between selected Tweets and the disparity between *all* of the Tweets in questions, using three different scoring schemes..

5.8.4 Methodology and Validation Remarks

In this section, the proposed improvements to the inference methodology have been implemented and assessed under a randomised controlled trial using Amazon's Mechanical Turk to crowdsource the validations.

Results from the analyses indicate the method's relative advantages over the techniques used in the previous chapter. In particular, the new method is applicable to generating appropriate interestingness decisions for Tweets from all users on Twitter, is capable of effectively *ranking* Tweets in order of interestingness, and is far more efficient in terms of training and supports 'on demand' predictions much more readily.

However, the crowdsourcing validations conducted were contributed to by people who shared no connection with the authors of the Tweets, and were thus assessing Tweets from outside of their own local network. Since it is known that users typically form followships between other users that produce information of interest, then this information is likely to be more relevant to users receiving the Tweets in these communities.

The following section addresses this area, in that the opinion of users assessing the Tweets within their own local networks is used to further the validations of the scoring mechanism.

5.9 Towards Addressing Information Relevance

In these analyses, results are studied from validations conducted through users assessing Tweets existing within their own local network. In particular, the interestingness scoring methodology will be validated against people's interestingness decisions on Tweets from those users they directly follow. Interactions with Twitter in this section relate to v1.1 of the Twitter API, as this research was conducted after the mandatory switch-over to this version.

Through assessment in this way, the Tweets being assessed are more relevant to their 'environments', which, in this case, consist of those users who would naturally also receive these Tweets and who are making the interestingness decisions based on their content.

5.9.1 Planning the Further Validations

As will become clear, no initial data collection is required for these analyses. Instead, users contributing to the crowdsourced analysis interacted almost directly with Twitter during the course of their assessments, which involved the studying of Tweets sent from the friends of the assessing user.

For this purpose, a web application was set up and ran from July to August 2013, which allowed visiting users to 'sign in' using their Twitter account through OAuth. As with v1, v1.1 of Twitter's REST API directly supports this kind of behaviour, and provides the authenticated application with access keys enabling it to interface with the API on the authenticating user's behalf. Applications registered on Twitter can have different levels of elevation - from read-only, in which Tweets, follower information, and so on, can be retrieved; to read and write, with which new Tweets can be posted for the user and new followships can be created. An advantage of using OAuth in this fashion is that each user has a separate rate limit associated with it, meaning that the application

could retrieve a lot of information, if necessary, yet without exceeding the rate limit afforded to it by the new policies of v1.1 of the API.

In this case, the web application was advertised through word of mouth and through OSNs, such as Facebook and Twitter itself, as well as through Mechanical Turk. In the latter case, a special link to the site was provided to MTWs, and a code was displayed to them on completion of the task, which they could enter into Mechanical Turk in order to be paid. Participants contributing from the word of mouth and OSN categories are defined as 'organic' participants. Since the analysis depended on users assessing Tweets from their Twitter friends, participants could only take part if they had had a Twitter account with at least 30 friends.

Figure 5.6: Advertising the validation site on Twitter.

After signing into the read-only application[3] and beginning the procedure, participants were faced with a series of ten Tweet timelines. The first consisted of the most recent 20 Tweets from the participant's home timeline, and the next nine consisted of user timelines of the participant's friends. Although the selection of friends for the nine user timelines was done at random, a slight bias was applied towards selecting friends with a higher follower count. Due to the nature of scale-free graphs, there are many vertices with few edges, and few with many edges. As such, in order to obtain a more even distribution of user influence, the weighting was necessary to ensure that the scoring mechanism could be validated against a range of users expressing a wider

[3]Twinterest: source available at https://github.com/flyingsparx/twinterestingness

variety of retweet counts.

Similar to the initial validations using Mechanical Turk in the previous section, participants were simply asked to select the Tweets that they found to be the most interesting from each of the timelines and were not able to proceed to the next timeline without selecting at least one Tweet. Note that, at this stage, the Tweets being assessed did not have interestingness scores applied to them. A Tweet in a timeline that was selected was considered to be interesting, and those not selected were non-interesting.

5.9.2 Assigning Scores to the Assessed Tweets

Through running the validation application, a total of 580 timelines were assessed, consisting of 389 contributed to by MTWs and 191 from organic participants. The totals are not precisely divisible by ten since not all participants assessed all of their ten timelines before leaving the application, but no one participant contributed more than ten timeline assessments. In this case, all responses were considered as confident since it was not appropriate under the conditions of the validation test to gain more than one assessment for each Tweet. Although there was likely some friend overlap between the participants, this was not necessarily the case in the vast majority of users assessed. In cases where there the same Tweet was assessed by more than one user the majority vote was chosen, weighted towards positive interestingness decisions in the case of ties.

The validation test resulted in a set of just under 10,000 Tweets, authored by 936 unique users, that had interestingness decisions made on them as part of the assessment of the participants' home timelines and friends' user timelines. These Tweets became the testing dataset T, and in order to determine their predicted expected retweet counts using trained classifiers as part of assigning the interestingness scores to each of the Tweets, two procedures were required to take place;

- Collect further data on each assessed *author* in order to generate the 'user' mod-

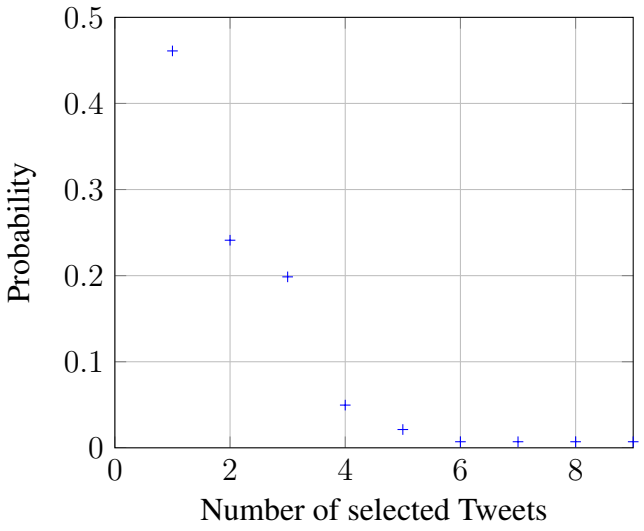


Figure 5.7: Probability of selecting different numbers of Tweets from each timeline.

els, and;

- Collect further data on each assessed *Tweet* in order to classify it against the global model and the relevant user model.

The global model used was the same large model generated during the previous validation tests.

For privacy reasons, each participant's Twitter API credentials were not maintained by the application and so standard authenticated REST API requests were performed to collect the additional data required. In particular, in August 2013, each of the 936 users representing $t.\text{author}_O \forall t \in T$ were queried under an identical collection scheme to that used as part of the previous validation; information on the author itself and on a sample of the author's followers and friends. The collected information was also assigned to each of that user's Tweets $t' \in T$ so that an instance could be built for every $t \in T$ according to the features described in Table 5.3. These Tweets were then classified by the global model and their appropriate user model, which was built from its author's features, in order to eventually produce the two scores.

It should be noted that if a particular user follows another whose account is protected (see earlier in the thesis for further information on this), then the former user's API credentials can be used to view the latter's information and Tweets. However, since, during the data-collection, a static account was used to query the API, then Tweets and user information for accounts that are protected could not successfully be retrieved. This means that user and Tweet data for these users could not be collected for the purposes of training the user model and testing Tweets against this and the global model, and thus Tweets from protected authors had to be removed from T. The numbers stated in this section are those of the *final* dataset after removing these Tweets and users.

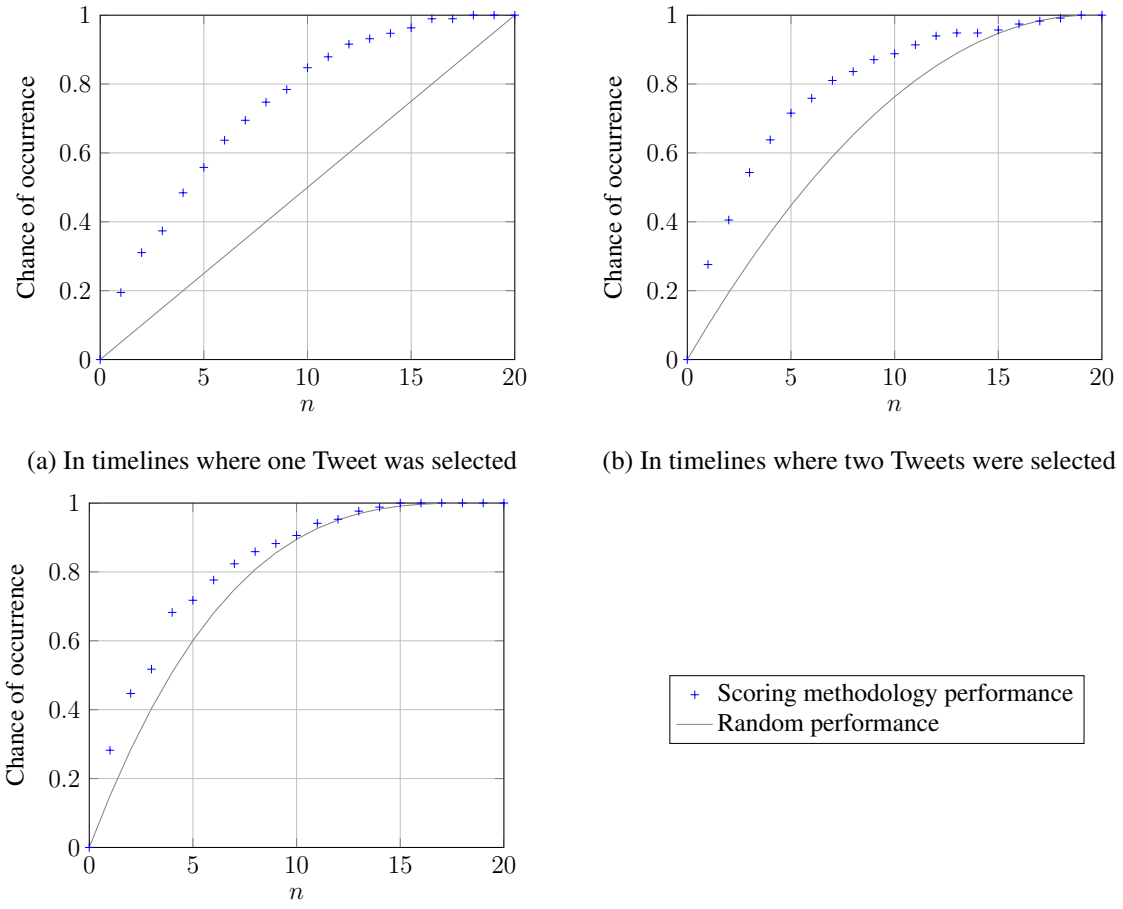
5.9.3 Results from the Further Validations

In this section, the patterns observed through the comparison of the Tweers inferred as interesting through the scores and those indicated as interesting by the human participants are analysed. There was no significant difference observed between the accuracy of the scores against the MTWs' and the organic participants' decisions, and thus the combination both sets was considered in the following analyses. The $s_G(t)$ was used as the scoring scheme for the analysis in this section.

Ranking performance in further depth

In the prevoius validation, the performance of the interestingness scores in ranking Tweets was assessed on a per-question basis. The same concept is expanded here to apply a similar assessment of the scores on the present validation test.

In this case, each assessed *timeline* was ranked in order of descending interestingness score in an effort to find the probability of a participant selecting a Tweet occurring in the top n of Tweets. Timelines were up to 20 Tweets long, compared to the five used in the Mechanical Turk questions in the initial validation test, but the scores have again demonstrated that they are effectively able to rank Tweets. Since the timelines



(a) In timelines where one Tweet was selected

(b) In timelines where two Tweets were selected

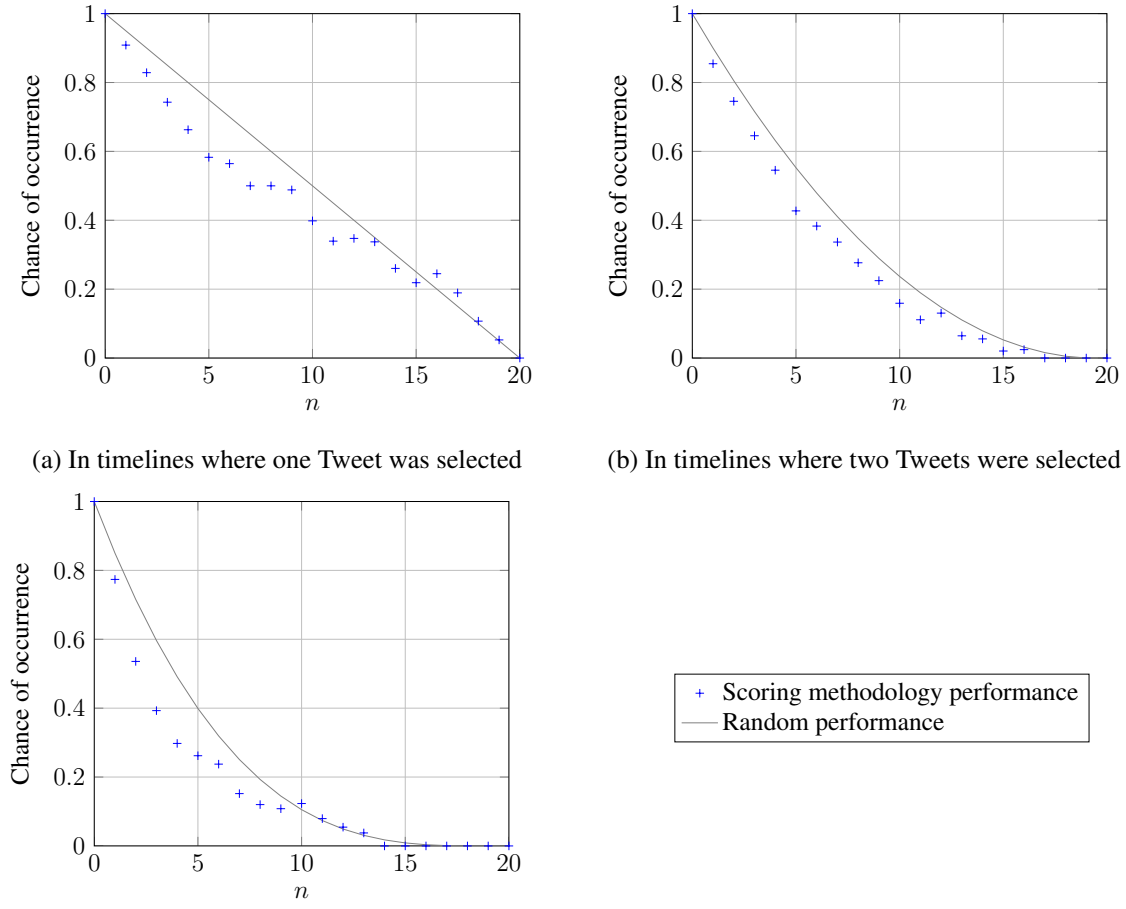
(c) In timelines where three Tweets were selected

Figure 5.8: The chance of a participant selecting one of the *highest* n ranked Tweets in the timeline.

are bigger than the questions used before, the chance of a participant selecting multiple Tweets from a timeline was greater, as indicated by Figure 5.7. To illustrate this, the results for this analysis are demonstrated against the appropriate random performance benchmark produced by the different selection criteria.

It is clear that the scores are able to identify interesting information from the noise around it, and so further analyses were conducted into the performance of the scores in detecting *un*-interesting information. In this scenario, each timeline had its Tweets ranked in order of *ascending* interestingness score and calculations were carried out into the probability of participants *not* selecting the *bottom* n interesting Tweets in

each timeline.



(a) In timelines where one Tweet was selected

(b) In timelines where two Tweets were selected

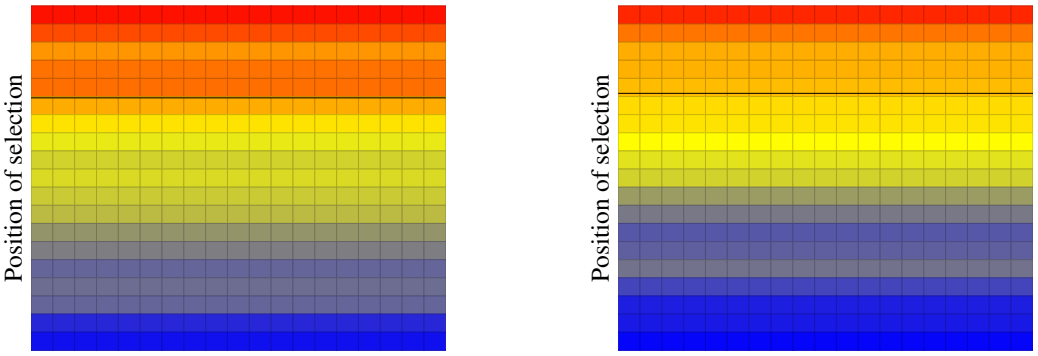
(c) In timelines where three Tweets were selected

Figure 5.9: The chance of a participant *not* selecting one of the *lowest* n ranked Tweets in the timeline.

Although the results showed that, with the different selection criteria, the scores are able to assist in identifying non-interesting information, the difference between this performance and the random selection case is not as significant as with detecting the positively interesting Tweets.

Crowdsourced timeline selections

A brief study was additionally made into the selections of Tweets made by the participants. Of particular interest is the *difference* in performance between those par-



(a) Selections made by organic participants

(b) Selections made by MTWs

Figure 5.10: Heatmaps illustrating the timeline position of selections made by participants. Mean selection position is indicated.

iticpants that were paid to take part (the MTWs) and those who took part without being paid (the organic participants), and whether one group was more likely to select Tweets near the top of the timeline without scrolling down to read and select those at the bottom of the timeline. Reasons for this case could be laziness on the behalf of the participant, or simply for speed.

The study revealed that there wasn't a significant difference between the two participant groups. The organic participants, on average, selected the Tweet at position 6.07 in the timeline, and the MTWs selected Tweets at the average position of 5.83 out of 20 maximum available positions. Whilst these selection position averages are both relatively near to the top of the timeline, it should be noted that the *mean* timeline length was of 14 Tweets, and thus purely average random selections should be made around the mark of the seventh Tweet.

It is felt, therefore, that there was some bias in both participant groups in that they were both slightly more likely to select Tweets nearer the top of the timeline than scroll down to view, and make interestingness judgments on those, nearer the bottom of the timelines. As with the previous validation tests, it was also possible to demonstrate that the maximum scores between Tweets in a particular timeline is greater in cases where only selection is made by the participants (Figure **??**).

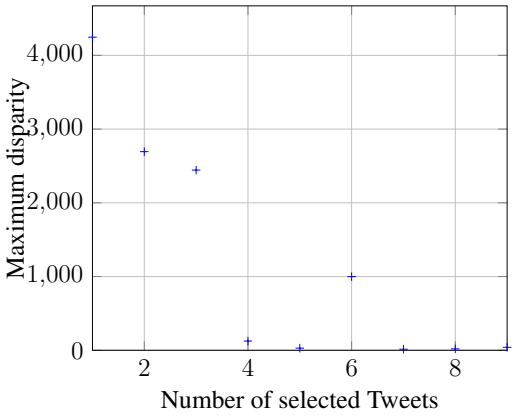


Figure 5.11: Relationship between the number of selected Tweets in a timeline and the maximum score disparity of the timeline.

5.10 Chapter Summary

In this chapter, an improvement over the previous iteration of the Tweet interestingness inference methodology has been introduced, tested, and analysed. Bringing the research into the social structure of Twitter forward from the previous chapter, it was possible to determine areas for improvement and the useful metrics for governing the selection of new features.

5.10.1 Interestingness Scores

The new methodology introduced the notion of scores, which can be assigned to Tweets in order to signify their relative interestingness. These scores are based on the ratio between the popularity of a Tweet, measured by its observed retweet count, and a value representing an *expected* retweet count for the Tweet. The scoring mechanism works such that different types of users, including across influence and activity frequency[4] ranges, can have their Tweets assessed on the same scale.

Two scoring schemes were set up, which are derived from distinct methods for gener-

[4]The activity frequency is how often a user posts Tweets

ating the expected retweet count. One method is based on comparing a Tweet's (and its author's) features to a *global* model trained on a large number of Tweets collected from Twitter. The other is generated through the comparison of the Tweet's features to a *user* model trained only on other Tweets posted by that particular user. Generally, there was found to be non-significant difference between the performance of the two scores, however, and thus they were both used interchangeably during the validations.

5.10.2 Methodology Validations

Two sets of validations were conducted into verifying the performance and accuracy of the new methodology and the scores it produced - one in which Tweets were placed into questions on Amazon's Mechanical Turk, in which MTWs were asked to select the most interesting Tweets; and another, in which users were asked to sign-in through Twitter and then assess Tweets from users they actually follow.

In the first case, the participants shared no connection with the authors of the Tweets they were assessing (except in the case of large coincidences), and were therefore assessing Tweets on a *global* interest level. In particular, this largely involves determining the interesting information from the noise around it. In the second set, information *relevance* came more into play, since participants were assessing Tweets from users they have already declared an interest in (through the action of following).

In both test cases, the validations showed the scores to be able to appropriately label Tweets according to interestingness in a variety of different ways. The second test included an analysis demonstrating that the scores are more efficient at determining interesting Tweets than *un*-interesting Tweets, the latter of which would be useful in deciding on a set of Tweets to discard from an interesting set.

5.10.3 Improvements and Qualities

The newly introduced methodology presents several improvements over that described in the previous chapter. In particular, the performance of the scores have shown a large accuracy improvement in determining interesting information. The previous method also did not take into account *how* interesting a particular Tweet may be, and was only able to make a binary interesting/uninteresting decision for each Tweet.

Another large improvement is the ability of the scoring method to be applied to a much wider range of Tweets. The previous method was realistically unable to assess Tweets from users with more than 300 or so followers due to data collection inefficiency, the time taken, and the computational complexity involved in simulating large user graphs. The new method can be used to assign scores to Tweets in a more "on demand" fashion, where only a small amount of informaiton for each Tweet is required in order to generate the features needed to predict the estimated retweet counts. The scores also allow Tweets from many different sources to be assessed on the same scoring scale, meaning that Tweets on a mixed timeline can be appropriately compared to one-another, as demonstrated by the second set of validations.

Chapter 6

Critical Assessment and Conclusions

This chapter includes an overview and assessment of the work conducted in this thesis, bringing together the ideas from the initial research and how these have helped in developing the methodologies introduced in later chapters. The validations from the methods are further assessed, followed by how the research forming them may be taken further in future project. Finally, an overview of the thesis in terms of its contributions are described.

6.1 Critical Analysis of Research and Results

Following is an analysis of the research carried out over the stages described in the main chapters of this thesis, from the initial research into retweeting and the social graph through to the interestingness inference methodologies explained at later stages.

6.1.1 Analysis of Initial Research

- What was the use of initial research?
- Are the results sensible?
- How have the results shaped the further research?

6.1.2 Analysis of Final Results

- Have methods been able to sensibly predict retweet volumes?
- Have methods sensibly inferred Tweet interestingness?
- What might have worked better?
- Which parts were useless?
- Which parts helped develop other areas of research which may have provided further avenues of research ideas?

6.2 Further and Future Work

How can this research be taken further in the future?

- Use previous results to predict how far a tweet is likely to be retweeted (for advertising purposes)
- Useful for detecting the kind of messages that are likely to travel further
- As well as providing an interest level, the systems also predict sensible estimations on retweet volumes.
- Perhaps useful for measuring the spread of rumours.

One route for this would be to try and infer a user's local network from a set of their immediate parameters, drawing on our earlier work suggesting that the Twitter network has th e properties of a scale-free small-world graph. Through studying graph patterns, it is possible to make sensible inferences on the edges and nodes of a user's local network based o n their follower count. From this, a graph edge density can be calculated, $d = \frac{|E|}{|N|(|N|-1)}$, for use in generating a scale-free network.

- remove links between users, do still receive the information - (future work?)

6.3 Conclusions

6.3.1 Summary

Summarise events and processes covered, reiterate what the point of the work was and how each part of the work covered relates to that.

6.3.2 Contributions

Restate the original contributions (from Introduction section). Explain the ways in which the work done relates to the projected contributions, that it is novel and useful.

Appendices

Source code, further diagrams, ideas, etc.