# Conversation Dynamics in Social Media - Anticipating Repliers in Online Conversations

Master's Thesis

at

Knowledge Management Institute
Graz University of Technology
A-8010 Graz, Austria


submitted by

**BSc. Johannes Schantl**

Supervisor: Dipl.-Ing. Dr.techn. Univ.-Doz. Markus Strohmaier
Advisor: Dipl.-Ing. Claudia Wagner, DI (FH) Rene Kaiser

01.10.2013

# Konversationsdynamik in Social Media - Vorhersage von Kommunikationspartnern in Online-Konversationen

Masterarbeit

am

Knowledge Management Institut
Technische Universität Graz
A-8010 Graz,Österreich


vorgelegt von

**BSc. Johannes Schantl**

Begutachter: Dipl.-Ing. Dr.techn. Univ.-Doz. Markus Strohmaier
Betreuerin: Dipl.-Ing. Claudia Wagner, DI (FH) Rene Kaiser
01.10.2013

# Abstract

This thesis sets out to explore to what extent repliers in online conversations can be predicted, and investigates different factors which influence who will reply to a message. Therefore, this work reports an empirical study on two different social media applications which are partly used for conversations.

Anticipating repliers and turn-taking behaviour is a fundamental challenge for mediated video communication systems. One underlying aim behind this work was to take first steps in exploring how knowledge about social media conversation behaviour could be utilized in such systems. The massive amounts of data that social media generates has facilitated the study of online conversations on a scale unimaginable a few years ago.

First, a significance analysis was conducted to evaluate the predictive power of topical, social and activity factors. Second, in a prediction experiment a binary classification model was set up which differentiates between users who will and users who will not reply to a certain message and gives insight into the predictability of reply partners in online conversations.

The results of this thesis suggest that in the case of Twitter conversations, social factors, which describe the strength of relations between users, are more useful than topical factors. This indicates that Twitter users' reply behavior is more influenced by social relations than by topical interests. The binary classification model achieves in the case of Twitter an F1-score of 0.74 when using social factors. Moreover, this thesis found that conversations in Boards.ie are dominated by users with high activity, and there is only a slight preference to reply to users with whom they have a social relation. Using only activity features in Boards.ie, an F1-score of 0.69 can be achieved.

The results presented in this thesis are relevant for researchers interested in understanding conversation dynamics in social media and system designers who want to enhance the user experience in online video communication tools.

# Kurzfassung

In dieser Arbeit wird untersucht, in wie weit nächste Kommunikationspartner in Online-Konversationen vorhersagbar sind. Des Weiteren wird der Einfluss unterschiedlicher Faktoren auf das Antwortverhalten von Benutzern untersucht. Zu diesem Zweck wurden empirische Experimente an zwei unterschiedlichen Social-Media-Anwendungen durchgeführt.

Die Vorhersage des Antwort- und Sprecherwechselverhaltens stellt zurzeit eine große Herausforderung in der Optimierung von computerunterstützten Video Kommunikationssystemen dar. Ein grundlegendes Ziel dieser Arbeit war es, die ersten Schritte in der Untersuchung, in wie weit Wissen über das Verhalten von Social-Media-Konversationen in diesen Systemen genutzt werden kann, durchzuführen. Erst die enorme Menge an Daten in Social-Media-Anwendungen machte komplexe Vorhersagestudien von Online-Konversationen möglich, welche vor ein paar Jahren noch undenkbar gewesen wären.

Das erste Experiment, die Signifikanz-Analyse, misst das Potential von sowohl thematischen, sozialen und Aktivitätsfaktoren, Antwortende auf eine bestimmte Nachricht vorherzusagen. In einem zweiten Experiment, dem Vorhersage Experiment, wird ein binäres Klassifikationsmodell verwendet, um Benutzer in zwei Klassen zu unterteilen: In die Klasse der Benutzer, die auf eine bestimmte Nachricht geantwortet haben, und in die Klasse der Benutzer, die dieselbe Nachricht gelesen haben, aber nicht geantwortet haben.

Die Ergebnisse dieser Arbeit zeigen, dass soziale Faktoren einen stärkeren Einfluss auf Twitter-Konversationen haben als thematische Faktoren. Das bedeutet, dass Benutzer in Twitter-Konversationen eher mit enger verbundenen Benutzern kommunizieren, als mit Benutzern, die ähnliche thematische Interessen haben. Das binäre Klassifikationsmodell erreicht im Fall von Twitter bei Verwendung von ausschließlich sozialen Faktoren einen F1-Score von 0,74. Im Gegensatz dazu werden Boards.ie Konversationen hauptsächlich von Benutzern mit starken Aktivitätsfaktoren dominiert. Soziale und thematische Faktoren haben in Boards.ie Konversationen nur einen sehr kleinen Einfluss. Bei ausschließlicher Verwendung von Aktivitätsfaktoren im Klassifikationsmodell wurde in Boards.ie ein F1-Score von 0,69 erzielt.

Die Ergebnisse in dieser Arbeit sind vor allem für Forscher, die sich für Konversations-Dynamik in Social-Media interessieren von Nutzen, aber auch für Systemdesigner von Online-Video-Kommunikations-Software, die versuchen, das Anwendererlebnis in diesen Systemen zu verbessern.

## Statutory Declaration

*I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.*

| | | |
|---|---|---|
| Place | Date | Signature |

## Eidesstattliche Erklärung

*Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.*

| | | |
|---|---|---|
| Ort | Datum | Unterschrift |

# Danksagung

Diese Masterarbeit wurde im Jahr 2013 am Knowledge Managment Institut der TU-Graz in Zusammenarbeit mit Joanneum Research in Graz durchgeführt.

Zu Beginn möchte ich ein herzliches Dankeschön an meinen Begutachter Prof. Dr. Markus Strohmaier richten. Er hat mich im Laufe einiger Vorlesungen für das Themengebiet WebScience und soziale Medien begeistert und viele Grundlagen gelehrt. Außerdem hat er auch sehr gutes Feedback zu dieser Arbeit gegeben.

Natürlich möchte ich mich auch bei meinen Betreuern Claudia Wagner und Rene Kaiser bedanken. Sie haben mich im Laufe dieser Arbeit sehr unterstützt und laufend Feedback und Tipps gegeben. Vor allem möchte ich mich für die Unterstützung bei der Verbesserung und Korrektur dieser Diplomarbeit bedanken.

Da diese Arbeit auch den Abschluß meines Masterstudiums darstellt, möchte ich mich bei all den Leuten bedanken die mich im Laufe des Studiums unterstützt haben. Dazu gehören natürlich meine Eltern die unendliche Geduld bewiesen haben und mich immer unterstützen. Außerdem möchte ich mich bei meiner Freundin Iveta, und auch bei meinen Freunden mit denen ich die letzten Jahre in derselben WG gewohnt habe, und die mich immer sehr inspirierten, bedanken. Vielen Dank!

Johannes Schantl
Graz, Austria, Oktober 2013

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

Recently, social media platforms like Twitter, Facebook and Google+ are gaining increased attention from millions of users all around the world. According to a study from Nielson [20], users spend more time on social media platforms than on any other websites. Nielson points out that in July 2012 users spent around 121.1 billion minutes using social media, an increase of 36% compared to July 2011. According to Alexa [1], a Web service for ranking the top used websites in the world, the social network Facebook is the second most popular website, whereas the microblogging platform Twitter is ranked as the eleventh most popular site on the Web.

In addition to sharing information with friends and other users, a main usage of social networks is to communicate with other people [14]. The massive amount of data produced by users on a daily basis provides the basis for data analysis done by researchers from different areas. Trending topics at popular Web science conferences such as the ICWSM, ACM Web Science or the ISWC focus, for instance, on social network analysis to identify communities or authorities, as well as on new text categorization and topic recognition methods, and also on analyzing sociological aspects in social networks such as the communication behavior in conversations, which is the main focus of this thesis.

This thesis gives insights into the dynamics of conversations. Especially, it focuses on finding inherent patterns in the reply behavior of users participating in social media conversations, in order to be able to predict if a user is likely to reply to a message or not. The need for this understanding emerges from a project aiming to improve Orchestration in group videoconferencing by using social media information from/about the members of the communication session to predict who will reply to the current speaker - cp. [15]. This task was simplified by predicting whether a user is likely to reply to a certain user or not. But the approach presented in this thesis can be easily extended to calculate the most likely replier. In addition to this specific purpose, finding patterns in online conversations

can also provide empirical tests of social theoretical models that have been proposed in the literature (see e.g. [18]).

When it comes to the theoretical study of online conversations, a natural assumption would be that the closer the friendship between two users A and B, the more likely user A replies to a message of user B and vice versa. Another hypotheses would be that conversations are driven by topical factors, and that the probability of user A replying to user B increases with their topical similarity - i.e., with the extent to which they talk about the same topics. Finally, one would assume that users who are in general more communicative are more likely to reply than other users. In this thesis the above mentioned natural hypothesis are evaluated in more detail on social media conversations. Therefore this work presents comprehensive insight into the fundamental question why users' are replying to each other.

In this work the predictability of users' reply behavior is measured on two different social media applications, the microblogging platform Twitter and the Irish bulletin board Boards.ie. A comprehensive set of features is proposed to quantify the major social and topical factors which may impact users' communication behavior. While topical factors capture the similarity of topical interests between users, social factors measure the strength of the relation between users. In addition to topical and social factors, also activity features which describe how active, how communicative and how popular a user is are added as covariates.

In order to explore the impact of topical, social and activity features on reply behavior, a significance analysis is performed which consists of statistical hypothesis tests and of a model-based significance test. While statistical hypothesis tests analyze whether a feature has an impact on the reply behavior or not, model based significance tests can further give insight into the magnitude of the features' predictive power. Furthermore, a binary classification model is used in the prediction experiment to measure the accuracy to predict whether a user is likely to reply to a message or not when using different sets of features (only topical, only social, only activity, all features). Hence, this experiment explores to what extent users who are replying to a message can be distinguished from users who are not replying to the same message. The better repliers can be distinguished from non-repliers the higher is the predictability of repliers in conversations.

## 1.2 Objective and Research Questions

One of the main objectives of this thesis is to analyze the dynamics of conversations on social media and specifically the predictability of future conversation partners. This includes finding an appropriate feature set, gathering a dataset for Twitter and Boards.ie, and developing a prediction model for repliers. In addition, this research gives detailed insight into the predictive power of different kinds of features in predicting repliers. The

main research questions of this thesis are:

- RQ1: To what extent is communication on social media influenced by social and topical factors?
- RQ2: To what extent are conversation partners on social media platforms predictable?

## 1.3 Contributions

The main contributions of this thesis are the following:

- A dataset was created which consists of 3850 Twitter conversations and gathered for 9122 users all recently published messages, their membership lists, their profile information and follower/followee network.
- This work defines a comprehensive set of features to quantify the major social and topical factors which may impact users' communication behavior. Further, it gives insights into the predictive power of each feature.
- In addition, a prediction experiment was conducted to analyze the predictability of reply candidates in conversations.

## 1.4 Structure of this Thesis

The next chapter describes the theoretical background to this thesis and also presents related research to this work. It includes a short introduction of the microblogging platform Twitter and the Irish bulletin board Boards.ie, a section about related research, and further covers the basics of logistic regression and highlights different aspects of topic extraction methods in documents. Chapter 3 explains in more detail the experimental setup of the prediction analysis. It covers the basic idea of our experiments, explains the dataset generation, introduces the feature set and gives a detailed explanation about different methods used in the experiments. The result of the Twitter and Boards.ie experiments are presented in Chapter 4. Chapter 5 discusses and summarizes the results of the research questions. Finally Chapter 6 concludes this thesis.

Parts of this thesis have been published in [23]:

- Johannes Schantl, Rene Kaiser, Claudia Wagner, and Markus Strohmaier [2013]. The utility of social and topical factors in anticipating repliers in Twitter conversations. In Proceedings of the 5th Annual ACM Web Science Conference, WebSci 13, pages 376-385.

# Chapter 2

# Related Work and Background

This chapter presents related research to this work and introduces the theoretical background of this thesis. Section 2.1 gives a short introduction about the social media applications Twitter and Boards.ie. In Section 2.2 a profound overview of related research is presented, focusing on Twitter research and on research which analyses communication behavior. Section 2.3 covers the basics of logistic regression, highlights some aspects of topic extraction in documents and presents three different topic extraction methods which are used in the experiments presented in this work.

## 2.1 Twitter and Boards.ie Introduction

### 2.1.1 Twitter

Twitter was launched in 2006 and is one of the most popular microblogging services in the world. Users may write short messages, called *Tweets*, which are limited to 140 characters. Information consumption on Twitter is mainly driven by explicitly defined social networks. That means, a user sees the messages authored by the users he/she follows on their Twitter timeline in reverse chronological order.

Tweets can be sent via e-mail, a Web browser, SMS, or third party applications and are displayed in the user profile.

#### Follower/Followee Network

A user $u_1$ is called *follower* of user $u_2$ if $u_1$ has established a follow relation with $u_2$ and, in the same example, user $u_2$ is a *followee* of user $u_1$. A user is called $u_3$ a *friend* of user $u_1$ if $u_1$ has established a follow relation with $u_3$ and vice versa.

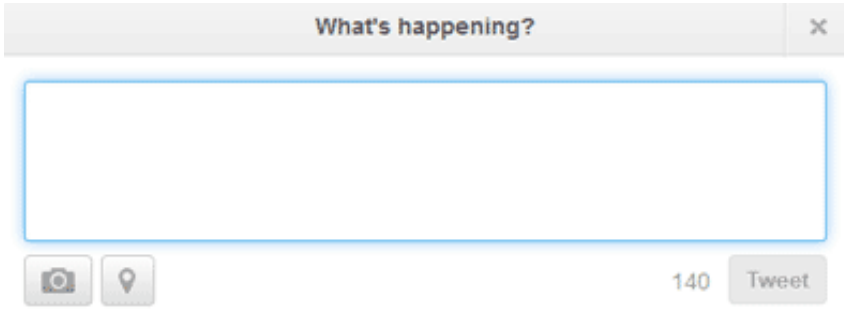


**Figure 2.1:** Screenshot of Twitter's Web user interface for writing Tweets.

### Mentions/Replies

Tweets can include so called mentions which consists of the "@" sign followed by a user name, e.g. "@user12". Mentions allow, as the name says, to mention other Twitter users within a Tweet, and can be included anywhere in the Text. All Tweets containing mentions are listed in the Mentions tab on Twitters Connect page.

When clicking on the "Reply Button" in Twitters Web interface, Twitter automatically adds a "@username" at the beginning of the Tweet, where *username* refers to the user who is replied. Therefore replies are using mentions as well.

### User Lists

A user list is a list of Twitter users which is created by a certain user. Users can create their own lists or subscribe to lists created by others. A list timeline shows a stream of weets from the users subscribed to this list. Lists are used to group users and to manage the stream of messages in a user's timeline. It is not possible to write a Tweet only to members of a list.

### Hashtags

A hashtag is used to mark special topics or keywords in a Tweet. To mark a word as a hashtag, one has to prefix the # symbol to the word. Hashtags in Twitter give users a way to categorize messages. In the Tweet below (example from [28]) the user @eddie added the hashtag #FF. FF stands shorthand for "Follow Friday," and is a weekly tradition where users recommend people whom others should follow.

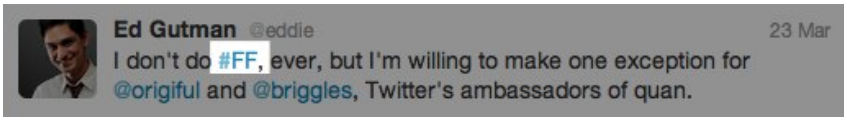


**Figure 2.2:** Example of hashtag usage.

#### Twitter API

Twitter offers the following three different Application Programming Interfaces (APIs) to crawl user information:

- REST API [26]: The Representational State Transfer API (REST API) covers different basic Twitter functions, such as posting a Tweet or Retweet and reading public user information like user profiles, follower and followee lists, membership lists and the like. At October 2012 the rate limit for using the REST API version 1.0 is 350 requests/hour for authenticated users and 150 requests/hour for unauthenticated users.
- Streaming API [27] : The streaming API offers the possibility to get real-time streams of Tweets. Twitter offers several different streaming endpoints like public streams, user streams and site streams. There is no public information available about Twitter's rate limit using the streaming API, but Twitter stated that the rate limit depends highly on the number of reconnections to the streaming endpoint from the client.
- Search API [29]: This API allows querying the real-time index of Tweets from the last 6 to 9 days. The Search API performs the same requests as the Twitter search page. The rate limit of this API is not officially known at the moment, but it was published by Twitter that the limit should be different from the limit of the REST API in version 1.0.

One important issue when using any Twitter API is to continuously check whether the request limit is already reached or not. It can happen that the user account gets blocked by Twitter when there are too many requests send to the Twitter API after reaching the request limit. The conversation crawler described in Section A.2.1 shows a simple mechanism for preventing this issue.

### 2.1.2 Boards.ie

Boards.ie is an Irish bulletin board founded in 1998 and according to Alexa [1] it belongs to the top 20 most popular websites in Ireland. There are a wide variety of topics discussed in Boards.ie, ranging from solution oriented discussion about technology issues

to discussions about trending topics in music, science, art, sport etc. Figure 2.3 shows the Web interface of Boards.ie. The top-level site document in Boards.ie, links to different top-level forums such as Arts, Sports or Technology. These top level forums link to sub-forums, for instance the Computer and Technology forum, and each sub forum contains threads which are used to ask questions or discuss with other users.


**Figure 2.3:** Web interface of the Irish bulletin board Boards.ie.

## 2.2 Related Research

### 2.2.1 Twitter Research

#### Twitter Usage Intention

In one of the first papers about *why we use Twitter*, Java et al. [14] analyses topological, geographical properties and usage intentions of Twitter users. They categorize the intentions of Twitter users into four different groups:

- Daily chatter: The most messages in Twitter refer to daily routines or to what people are doing at the moment. Daily chatters are the most common users in Twitter.

- Conversations: Approximately every eight Twitter message contains a conversation, and 21% of users use conversations.

- Sharing Information/URLs: 13% of all messages contain URL's. Due to the limit of 140 signs per Tweet, services like TinyURL were used to shorten URL's.
- Reporting news: Twitter is often used for reporting news.

### Twitter Dialogs

In a research from 2012 which investigates the structure of Twitter dialogs in more detail, Macskassy et al.[17] evaluated the following questions:

- How do users behave in terms of the amount of Tweets appearing in dialogs compared to other Tweet types like mentions, Retweets and Tweets?
- How do dialogs look like in terms of number of participants, the length of dialogs and whether some users are more dominant in dialogs than they users?

The dataset they are using for this evaluation contains Tweets from 2400 users from the Middle East within a time period of one month.

They found out that 13% of the user activity focuses on dialogs. Further 42% of users did not participate in dialogs at all, 92% of all dialogs were between two people and the average number of messages in dialogs is less than 5 Tweets. They also identified dominant users in dialogs by calculating the dominance distribution in those dialogs.

### Usage of @ sign

Honeycutt et al. [11], analyzed the usage of the @ sign in Twitter in 2009. They set up several research questions. In relation to this thesis the following questions are important:

- What is the usage of the @ sign in English Tweets?
- Is there a difference in the content of Tweets with @ sign and without?
- How long are interactive dialogs, and does the length differ in dialogs with @ sign?

Their dataset contains around 37000 Tweets crawled from Twitter's public timeline in four different one hour intervals. They found out that around 91% of all Tweets use the @ sign for addressing other users and around 5.5% use the @ sign for referencing another person. Other usages of the @ sign are emails locating to a position etc. They found out that the time span of a conversation duration ranges from 25 seconds to around 54 minutes, the average conversation length is 4.6 Tweets and the average number of participants is 2.5. Table 2.1 lists the usage statistics of different Tweet types with/without

the @ sign. In table 2.1 one can see the usage statistics of the @ sign among the different Tweet types.

| | **without/@** | **with/@** | **total** |
|---|---|---|---|
| about addressee | 21(33%) | 2(1%) | 23(11%) |
| announce/advertise | 0(0%) | 14(10%) | 14(7%) |
| exhort | 7(11%) | 1(1%) | 8(4%) |
| info for others | 10(16%) | 1(1%) | 11(5%) |
| info for self | 2(3%) | 9(6%) | 11(5%) |
| meta-commentary | 0(0%) | 4(3%) | 4(2%) |
| media use | 4(6%) | 14(10%) | 18(9%) |
| express opinion | 5(8%) | 8(6%) | 13(6%) |
| other's experience | 1(2%) | 10(7%) | 11(5%) |
| self-experience | 11(17%) | 73(51%) | 84(41%) |
| solicit info | 0(0%) | 3(2%) | 3(1%) |
| Other misc. | 2(3%) | 5(3%) | 7(3%) |
| Total | 63 | 144 | 207 |

**Table 2.1:** Usage statistics [11] of the @sign for different Twitter message types.

### Types of Tweets

Naaman et al. describes in [19] a content-based categorization system for messages written in Twitter and further analyzed the usage activity for all types of messages.

Table 2.2 shows the categorisation scheme of Tweets set up by Naaman et al., while figure 2.4 shows the usage statistics of these types.

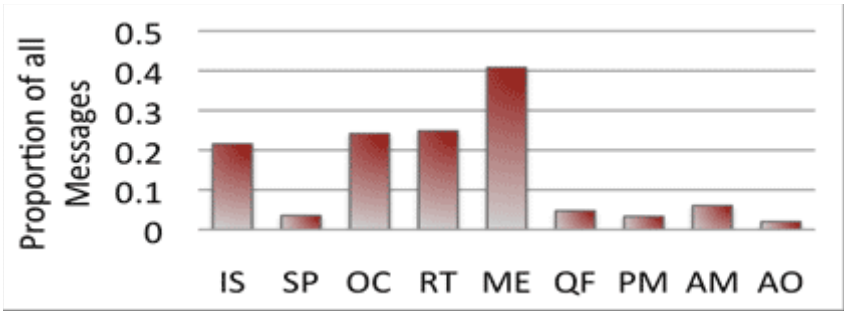


**Figure 2.4:** Usage statistics [19] of the different categorized message types in Twitter.

| Type | Examples |
|---|---|
| Information Sharing (IS) | ”15 Uses of Wordpress” |
| Self-Promotion (SP) | ”Check out my blog I updated 2day 2learn abt tuna!” |
| Opinions/Complaints (OC) | ”Illmatic is greatest rap album ever” |
| Random Thoughts (RT) | ”The sky is blue in the winter here” |
| Me now (ME) | ”tired and upset” |
| Question to followers (QF) | ”what should my video be about?” |
| Presence Maintenance(PM) | ”gudmorning twits” |
| Anecdote (me) (AM) | ”oh yes, I won an electric steamboat machine and a steam iron at the block party lucky draw this morning!” |
| Anecdote (others) (AO) | ”Most surprised @user dragging himself up pre 7am to ride his bike!” |

**Table 2.2:** Twitter messages categorisation scheme defined by Naaman et al. [19].

### Follower Intention

Rowe. et al. [22] analyzed the follower intention of users using the KDD Cup dataset. Similar to Twitter it uses a follower/followee relation in the social network. They base their analysis on three different types of features: social, topical and visibility features.

- Social: Consists of the mutual followers, mutual followees, mutual friends count and mutual neighbors between two users.
- Topical: For calculating the topical similarity, they are using user tags and their associated concepts.
- Visibility: Consists of features which measure the visibility of the prospective followee to a user. For this they are using the retweet count, mention count, comment count, weighted retweet count, weighted mention count and weighted comment count.

To determine whether the intention of user A to follow user B was topically or socially driven they were comparing logistic regression models based on different user-follower sets. The first set contains a 10% random sample within the whole dataset. They found out that the user’s intention to follow is generally more topically driven then social. For the other user-followers sets they are binning different users-follower pairs together based on a user’s topical focus and social connectivity. In this way they found out that users who have a low topical focus base their follower decision on social rather than topical features, and users which are highly socially connected base their decision also on social rather than topical features.

#### Friends Network

In [13], Hubermann et al. analyzed the structure of the followers network. They found out that Twitter users have a very small number of friends (users who have sent at least two direct messages to each other) in comparison to their number of followers/followees.Therefore, in addition to the very dense follower/followee network Hubermann et al. identified a sparse network, consisting only of friends.

### 2.2.2 Community Extraction

In [32], Zhang et al. found a method to successfully identify communities in Twitter based on the interests of users. Their approach was to first compute user similarity leveraging both textual contents and social structure. These features include URLs, Tweet content, hashtags, retweeting relationship and following relationship. All of these features are closely correlated with users' interests. To discover communities they further used user similarity measures as well as classical clustering algorithms.

### 2.2.3 Hashtag Recommendation

In an research from Yang et al. [31], they analyzed the dual role of hashtags usage in Twitter. They found out that a hashtag is used as a social bookmark that annotates content shared to other users, and assembles the folksonomy. Moreover, hashtags are used to mark the membership to a virtual community in Twitter.

#### Tie Strength

In a research from 2009, Gilbert et. al. [8] set up a model which classifies user pairs in strong and weak ties. While strong ties refer to trusted people of a user such as family members and friends, weak ties refer to merely acquaintances providing merely access to information. They were analyzing 2000 social media ties within Facebook and distinguish strong ties from weak ties with an accuracy of 85%. Gilbert et. al. suggest the following seven dimensions to calculate the tie strength:

- Intensity (wall words exchanged, inbox messages exchanged, etc.),
- Intimacy (days since last communication, appearance together in photo, etc.),
- Duration (days since first communication),
- Reciprocal services (links exchanged by wall posts, etc.),
- Structural (number of mutual friends, etc.),

- Emotional support (wall and inbox positive emotion words, etc.),
- Social distance (age difference, educational difference, etc.).

In addition, they were using a linear combination of all features to calculate the tie strength. The best dimensions to classify users in strong or weak connections were intimacy, intensity, duration and social distance. Later Gilbert matched the same approach to calculate the tie strength between users on Twitter [7], and analyzed how good this approach is applicable to other social media platforms next to Facebook. Referring to Twitter the following features are used to calculate the tie strength:

- Days since last communication,
- Days since first communication,
- Reply words exchanged,
- Mean tie strength of mutual friends,
- Follower difference,
- Shared links.

To evaluate the prediction accuracy in Twitter, Gilbert developed a tool called ”we meddle”. Using this tool Twitter users were able to classify their followers into strong and weak connections by hand. Further users can evaluate the classification results. As a result, they get as an upper bound of the misclassification rate of 15.7% from around 14000 examples. The research shows that the approach for calculating the tie strength in Facebook can be successfully applied as well for the social network Twitter.

### 2.2.4 Conversation Analysis Research

**Recommending Conversations in Twitter**

The paper [4] deals with recommending conversations to users. Their recommendation algorithm is based on three different kinds of features:

- The thread length (number of Tweets within a conversation).
- The topical relevance between user and conversation.
- The tie strength between a specified user and the user participating in the conversation.

To recommend a conversation to Alice, Chen et al. first collect all conversations from Alice's followees. Then they apply ranking algorithms to these conversations with varying features and display the highest ranked conversation to Alice.

To calculate the topical relevance the content of all messages from Alice and the messages within the conversation are represented as a TF-IDF weighted bag of words vector. The words in the weighted bag for the conversation are further enriched using the Yahoo Boss engine. For the calculation of the tie strength between two users, Chen et al. was mainly inspired by Gilbert [8]. The tie strength is mainly influenced by three factors, the existence of former direct communication, the frequency of such direct conversations and the tie strength of their mutual friends.

Moreover, they were making an on-line study to evaluate six different conversation prediction algorithms: A random prediction, prediction based only on the thread length, prediction based only on topical similarity features, prediction based only on the tie strength, prediction based on the sum of the tie strength and finally a prediction based on both the topical similarity and the tie strength. They evaluate these algorithms for different kind of user groups based on their usage intention (social or topical).

As a result, they found out that the performance of the algorithm is highly dependent on the Twitter usage intentions. Social intended users are more interested in conversations with participants of higher tie strength to the user. All algorithms performed better than the random baseline algorithm.

### Predicting Repliers

In [3] the authors explore the problem of predicting directed communication intention between Twitter users who did not communicate with each other before. Thus, the authors use various network and content features and conduct a link prediction experiment to assess the predictive power of those features. They set up a directed reply graph containing direct links between users who have replied in past. To calculate the user similarity they are using a combination of network and content proximity features. Further, they set up several combinations and weighting schemas for the features and randomly deleted links in the reply graph. In the next step they were trying to predict links between users within the graph to evaluate the different weighting schemes. In this research they are presenting an interesting approach to find interesting people to initiate communication with.

Compared to Chelmis et al. who were focusing in their research on predicting new communication links among users, the main focus in this thesis lies on the evaluation of different features in their power to predict if someone will reply to an author of a message or not.

## 2.3 Theoretical Background

### 2.3.1 Logistic Regression Model

#### Introduction

In areas such as Statistics and Machine Learning, regression methods have become a very important part in describing the relationship between an output variable and one or several input variables. In many applications the output variable is binary, referring to dichotomous pairs like success or failure, true or false, pass or fail etc. For these kind of output variables, the logistic regression model has become the standard method [12]. The goal of a regression analysis is to find the best fitting model to describe the relation between output variable and the input variables, and gain further knowledge about the influence of each single input variable to the output. In literature the input variables often refer to independent variables, predictors, covariates, or especially in Machine Learning tasks to features. Output variables are often called dependent, outcome or response variables.

While in linear regression models the dependent variable is always numeric, in logistic regression the outcome is categorical. Categorical variables can contain one or several classes [12]. The outcome is binary if only one class exists. While the binary logistic regression is only capable to handle binary outcome variables, there exists also an extension of this model which is capable to deal with multiclass dependent variables. This is known as multiclass logistic regression. As input variables, logistic regression can deal with categorical and numerical variables. In logistic regression these input variables are mapped to probabilities ranging from zero to one, using a logistic distribution function. There have been several functions proposed for use in the analysis of binary outcome variables, but there are some major advantages in the use of the logistic function compared to other functions. First, from a mathematical point of view it is a very flexible and easily used function. Second, the results of the coefficients are easy to interpret. The logistic function is defined as follows [12]:

$$\pi(x) = \frac{e^{\beta_0+\beta_1*x}}{1+e^{\beta_0+\beta_1*x}} \tag{2.1}$$

where $\beta$ denotes the coefficients of the model, x refer to the input and $\pi(x) = E(Y|x)$ represents the conditional mean of the outcome Y given x. The coefficients $\beta$ of the model refer to log odds ratios, which define the increase of the log odds when the input value $x$ is increased by one. The coefficients $\beta$ can be easily transformed to odds ratios using the inverse function of the $ln$, the $exp$ function.

This logistic function can be easily transformed into a logit function $g(x)$ which is linear and has an outcome ranging from $-\infty$ to $+\infty$. Therefore *g(x)* has all the desirable

properties of a linear regression model. The logit function is defined as follows[12]:

$$g(x) = ln * [\frac{\pi(x)}{1 - \pi(x)}] = \beta_0 + \beta * x_1 \tag{2.2}$$

The logit function refers to the logarithm of the odds, which describe the ratio of success to failure at a certain input $x$. For example an odds value of 2 means that the chance of a success is two times higher than a failure.

The following example from [12] aims to explore the usage of an logistic regression model. The example should find out the relation between the Coronary Heart disease(CHD) and the age of a person. The dataset contains hundred samples, containing the age of the person and whether the person has CHD or not. When drawing a scatter plot (each sample refers to a point in the plot) 2.5, it is difficult to see the functional relationship between age and CHD because the variability of CHD at all ages is big.

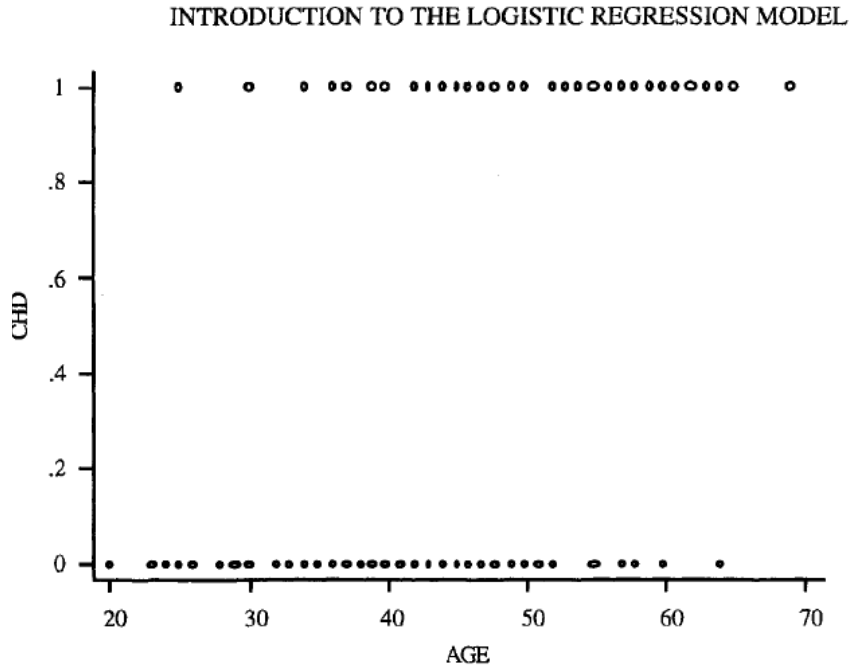


**Figure 2.5:** Scatterplot of CHD by age for 100 samples [12].

A common method to remove some variation in the data without removing the structure of the relationship between input and its outcome is to divide the input variable into several intervals and calculate the mean of the outcome for each interval [12]. Figure 2.6 shows for each age interval the occurrence of CHD in percent. After removing the variation a clearer picture of the relationship emerges, and it seems that with increasing age the proportion of people with CHD increases. The plot in 2.6 can be easily mapped by a logistic function, because with increasing age the percentage of CHD and further

also the odds (CHD=1/CHD=0) for each interval increase almost linear.

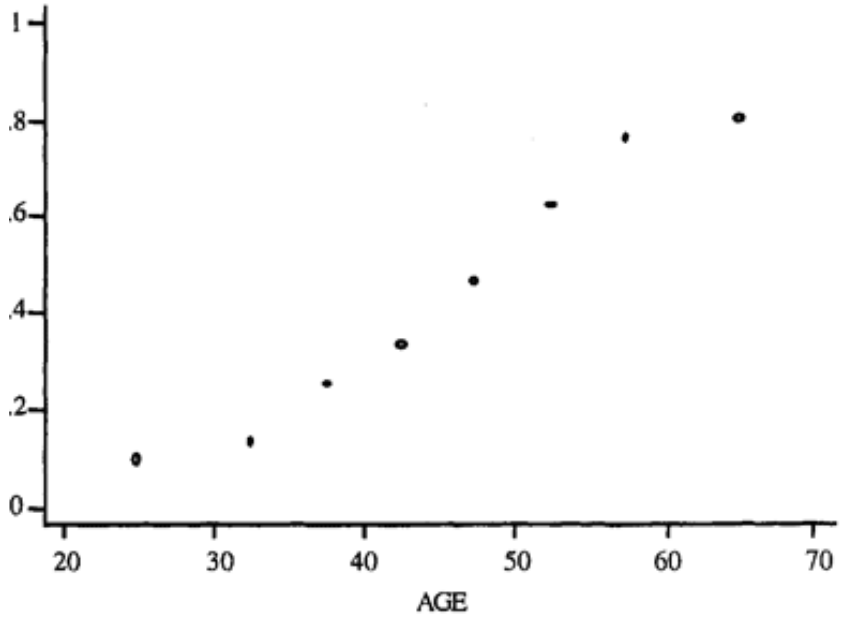


**Figure 2.6:** Frequency table of all age intervals by percent of CHD cases [12].

To estimate the coefficients of the logistic regression model, a maximum likelihood method is used. This method tries to find those values of the coefficients which maximize the probability of obtaining the observed data using the logistic regression model. In order to apply the maximum likelihood method, a likelihood function is constructed which calculates the probability to obtain the observed data as a function of the input variables. The likelihood function used in logistic regression is defined as follows [12]:

$$g(x) = \prod_{i=1}^{n} \pi(x)^{y_i} * [1 - \pi(x)^{1-y_i}] \tag{2.3}$$

where $y_i$ refers to output for a single input x. The likelihood function has a value of 1 when the observed outcome is perfectly fitted. A perfect fitted model is called saturated model. While in linear regression the likelihood function is linear and therefore the equations to get the coefficients can be easily computed, the likelihood functions for logistic regression is non-linear and require special methods for estimating the coefficients maximizing the likelihood function. A common approach to maximize the likelihood function is to use an iterative method, like the iterative weighted least square method.

#### Significance tests of the coefficients

After estimating the coefficients $\beta$ of the logistic regression model, one is usually interested in the significance of each input variable within the model. Significance in this

case means whether an input variable improves the performance of the model or not. One method to test the significance of an input variable relates to the question, whether a model which includes an input variable fit better to the observed data, then a model which do not include the variable. The answer of this question is to observe the difference of model fit with the variable compared to the model fit without the variable. A measure of how good the model fit gives the aforementioned likelihood function which value increases the better the model fits the observed data. A measure to calculate the model fit is the likelihood ratio D, which is defined as follows[12]:

$$D = -2 * ln(\frac{likelihood\_of\_the\_fitted\_model}{likelihood\_of\_the\_saturated\_model}) \tag{2.4}$$

The likelihood of the saturated model is always 1 therefore this formula can be simplified to the following formula:

$$D = -2 * ln(likelihood\_of\_the\_fitted\_model) \tag{2.5}$$

Further, to calculate the significance of a coefficient $\beta$ the difference between the deviance of the model with the coefficient and the deviance of the model without the coefficient has to be calculated, like the following[12]:

$$G = -2 * ln(\frac{likelihood\_without\_the\_variable}{likelihood\_with\_the\_variable}) \tag{2.6}$$

where G follows a chi-square distribution with 1 degree of freedom under the hypothesis that the tested coefficient is zero. Further one can look in chi-square table to gather the p-value which gives insight whether the difference is significant or not. This method refers to the likelihood ratio test.

Other approaches to test the significance of coefficients are the Wald Test and the Score Test, referring to [12] for further information.

### Interpretation of the Results

After fitting the logistic regression model to a provided dataset, usually one wants to draw practical conclusions from the estimated coefficients according to the research questions. The coefficients can be separated in the intercept and the coefficients for the input variables. The intercept $\beta$ represents the bias of the outcome, when all other coefficients are zero. In most cases the intercept is not of interest. The coefficients for the input variables represent the change in the outcome per one unit change in the input variable. These coefficients are log odds ratios as already mentioned earlier. Further, the interpretation of dichotomous and continuous input variables in the case of univariate

logistic regression models are described shortly. In addition, the interpretation of the coefficients in multivariate models is discussed, especially the confounding analysis and the collinearity issue.

In the case of binary input variables, it is assumed that the value for $x$ is decoded as 1 and 0. The coefficient can be calculated in the case of univariate logistic regression as the ratio of the log odds when x equals 1 to the log odds when the x equals zero. A positive coefficient value describe an increase in the probability of having an outcome value of 1. A coefficient with a value of 0 means that the probability of having an outcome value of 1 is 50% and therefore the related input variable would not improve to fit the data.

When a logistic regression model contains a continuous input variable, the interpretation is similar like in the dichotomous case. One assumption which is often made when using continuous variables is that the input variable are linear in the logit function. When it comes to non linearity, there are several methods to deal with this issue. One way is to separate the variable in several dummy variables. It can also be considered to use, for instance, high order terms like $x^2$ or other nonlinear scaling methods for the input variable to deal with the aforementioned issue.

In some studies containing multiple input variables it is also important to check for confounders. Imagine one can find, for example, a statistical influence of eating ice cream to drowning in the sea. One can suggest that there exists a causal relation between these two variables. A more likely explanation is that there exists a third variable, a so called confounder, for example, season which is effecting both variables, eating ice cream and drowning in the sea, and this would lead to the more realistic interpretation that during summer the people eat more ice cream and also drown more often in the sea than in winter.

A confounder is defined as a variable which has a significant effect on the outcome and is correlating with another input variable. A common method to check for confounding of an input variable B in a model with for example one input variable A, is to compare the estimated coefficients of A with and without B in the model. If there is a significant change in the value of the estimate of A than B is likely to be confounder. In some situations it is common to add potential confounder variables to see whether the effect of an input variable to the outcome remains when controlling or adjusting for these confounder's.

Another problem we want to mention here is an extreme case of collinearity (multicollinearity), where two or more variables are correlating very strongly with each other. This issue can run the maximum likelihood solver into numerical problems and delivers unstable results. A common way to deal with this issue is either to remove one of the collinear features or combining them.

For detailed information about logistic regression it is referred to the book "Applied Logistic Regression"[12].

### 2.3.2 Topic Extraction Methods

Topic Extraction in this thesis refers to the problem of finding terms in messages of users which reflect topical interests of a user. This is a common task in the area of text mining. Per definition, a text mining system is any system that analyzes large amounts of text and detects usage patterns to extract useful information [24].

This section will describe three different topic extraction methods applied in this thesis in more detail. The first method uses a Part-of-Speech-Tagger which is optimized for Twitter messages. The second and third method use both the AlchemyAPI[1] but differ in the applied extraction method (keyword extraction and concept tagging).

#### Part-of-Speech Tagging and Phrase Chunking

All three methods are based on the extraction of Part-of-Speech Tags. The process of classifying words into their word classes, also called parts of speech, is known as Part-of-Speech tagging, POS-tagging, or simply tagging[2]. Table 2.3 shows a commonly used Part of Speech Tagset. In relation to topic extraction, nouns(N) and proper nouns(NP) are especially important. Per definition proper nouns are nouns which refer to an unique entity such as Barcelona, Jupiter or Pluto.

After annotating each word of a text with an POS tag, one can use phrase chunking methods to recover phrases such as noun phrases and verb phrases. For instance, in the sentence *Michael Jackson is my favorite musician*, there exist a proper noun phrase Michael Jackson. Noun phrases are, for instance, verb-noun, noun-noun and adjective-noun combinations.

#### Common issues

The following three sentences illustrate some important issues when dealing with topic extraction.

1. I like football, basketball and watching TV.
2. Apache is my favorite helicopter.
3. Michael Jackson is my favorite musician.

A common approach to find topics in documents is to extract nouns from sentences. In the first example the nouns are football, basketball and TV. A limit of this method is, for instance, in the case of TV, the verb *watching* is neglected. The phrase *watching TV* is more specific than TV, and would give a better insight into the interests of a user.

[1] http://www.alchemyapi.com/

| Tag | Meaning | Examples |
|---|---|---|
| ADJ | adjective | new, good, high, special, big, local |
| ADV | adverb | really, already, still, early, now |
| CNJ | conjunction | and, or, but, if, while, although |
| DET | determiner | the, a, some, most, every, no |
| EX | existential | there, there's |
| FW | foreign word | dolce, ersatz, esprit, quo, maitre |
| MOD | modal verb | will, can, would, may, must, should |
| N | noun | year, home, costs, time, education |
| NP | proper noun | Alison, Africa, April, Washington |
| NUM | number | twenty-four, fourth, 1991, 14:24 |
| PRO | pronoun | he, their, her, its, my, I, us |
| P | preposition | on, of, at, with, by, into, under |
| TO | the word to | to |
| UH | interjection | ah, bang, ha, whee, hmpf, oops |
| V | verb | is, has, get, do, make, see, run |
| VD | past tense | said, took, told, made, asked |
| VG | present participle | making, going, playing, working |
| VN | past participle | given, taken, begun, sung |
| WH | wh determiner | who, which, when, what, where, how |

**Table 2.3:** Commonly used Part-of-Speech Tagset [2].

Whether more specific or more general topics are appropriate, depends on the application. A solution to get more specific terms would be to extract noun phrases instead of nouns.

Another common issue when extracting concepts is to deal with so called *Homonyms*. In the second example, Apache can on the one hand refer to the browser Apache but also the helicopter like in the sentence above. A *Homonym* is per definition a word which has different meanings. Contrarily different words with the same meaning, such as scream and yell, are called *Synonyms*.

When extracting common nouns from sentence three one would extract three words: Michael, Jackson and musician. In this sentence it would make more sense to gather Michael Jackson as one word. Michael Jackson is a so called proper name, which belong to the class of noun-phrases. To gather proper names one has to extend the Part-of-Speech tags by using phrase chunk taggers. These taggers are able to search for phrase patterns in the Part-of-Speech annotated text and apply a new label to the chunk of text.

### Twitter Part-of-Speech Tagger

The first method this thesis used for extracting topics is a Part-of-Speech Tagger optimized for Twitter messages developed by Gilbert et al. [9]. While common POS Taggers such as the Stanford Log-linear Part-Of-Speech Tagger [25] perform very well in formally written text (around 97% accuracy), they do not deliver sufficient results for documents containing abbreviations, emotions, retweets, hashtags and not grammatically correct text. Gilbert et al. developed a new POS Tagset [9] illustrated in table 2.4, which covers the requirements for these kind of documents.

The Twitter POS tagger reaches an overall tagging accuracy of 90%. Proper nouns are recognized with a recall of 71% and are often tagged as common nouns(N). The tagger is implemented in Java and includes an executable .jar file.

Next, the results from the Part-of-Speech tagger for different example sentences are shown. In the first examples, the result includes only proper nouns while in the second common nouns and proper nouns are extracted. Extracting proper nouns from the example sentences:

- I like football, basketball and watching TV.
  *Result: -*
- Michael Jackson is my favorite musician.
  *Result: Michael, Jackson*
- I am going on holidays to Barcelona.
  *Result: Barcelona*
- Ikr[2] smh[3] he asked fir[4] yo[5] last name so he can add u[6] on fb[7] lololol
  *Result: fb*

Extracting proper nouns and common nouns from the example sentences:

- I like football, basketball and watching TV.
  *Result: football, basketball, TV*
- Michael Jackson is my favorite musician.
  *Result: Michael, Jackson, musician*

[2] ”Ikr” means ”I know right”
[3] ”smh” means ”somehow”
[4] ”fir” is a misspelling or spelling variant of the preposition for.
[5] ”yo” is being used as equivalent to ”your”
[6] ”u” is an shortcut for ”you”
[7] ”fb” is an shortcut for ”facebook”

- I am going on holidays to Barcelona.
  *Result: holidays, Barcelona*

- Ikr smh he asked fir yo last name so he can add u on fb lolololol
  *Result: name, fb*

The result shows that in the above examples proper nouns as well as common nouns are extracted correctly by the tagger. Further, to be able to annotate proper phrases such as Michael Jackson a, the tagger has to be extended by phrase chunking methods.

#### Alchemy API

The Alchemy API[8] can be used to extract several kind of information from text like keywords, abstract concepts, named entities etc., and provides API calls for the different methods. Alchemy limits their free API usage (Oct 2012) to 1000 calls/day, but in case of academic use, the limit can be raised to 30000 calls/day . The Alchemy API supports more than a half-dozen languages including English, German, Spanish, Russian etc. The API can also deal with short messages like Tweets but to identify the language the message should have at least 20 characters. For shorter messages English is used as default language.

There are SDK's available for different programming languages such as Python, C++ or Java. Further Alchemy offers a command line tool for Linux/Unix systems

### 2.3.3 Alchemy API Keyword Extraction

The keyword extraction API call from Alchemy[9] extracts keywords from text. The keyword extraction works for 8 different languages such as English, French, German, Italian, Portuguese, Russian, Spanish and Swedish. As a response this API call delivers all keywords and its relevance within a document and optionally also makes a sentiment analysis for each keyword. The XML response for this API call has the following structure [10]:

```
1 <results>
2     <status>REQUEST_STATUS</status>
3     <url>DOCUMENT_URL</url>
4     <language>DOCUMENT_LANGUAGE</language>
5     <text>DOCUMENT_TEXT</text>
6     <keywords>
```

[8] http://www.alchemyapi.com/
[9] http://www.alchemyapi.com/api/keyword/proc.html
[10] http://www.alchemyapi.com/api/keyword/htmlc.html

```
7           <keyword>
8               <text>DETECTED_KEYWORD</text>
9               <relevance>DETECTED_RELEVANCE</relevance>
10              <sentiment>
11                  <type>SENTIMENT_LABEL</type>
12                  <score>SENTIMENT_SCORE</score>
13                  <mixed>SENTIMENT_MIXED</mixed>
14              </sentiment>
15          </keyword>
16      </keywords>
17  </results>
```

The keyword extraction functionality delivers the following keywords for these sentences:

- I like football, basketball and watching TV.
  *Result: football, basketball, TV*

- Michael Jackson is my favorite musician.
  *Result: Michael Jackson, favorite musician*

- I am going on holidays to Barcelona.
  *Result: holidays, Barcelona*

- Ikr[11] smh[12] he asked fir[13] yo[14] last name so he can add u[15] on fb[16] lololol
  *Result: ikr, smh, fb lololol*

There is officially no information available from AlchemyAPI about the internal methods used for each API call, therefore, the following statements are conclusions from the results in the above sentences. In these examples one can see that the keyword extraction extracts common nouns (football, TV,..), proper names (Michael Jackson) and also adjective-noun pairs, but the result does not include verb-noun pairs. In the last example which includes misspellings, abbreviations and interjections, the keyword extraction delivers not useful results.

[11] "Ikr" means "I know right"
[12] "smh" means "somehow"
[13] "fir" is a misspelling or spelling variant of the preposition for.
[14] "yo" is being used as equivalent to "your"
[15] "u" is an shortcut for "you"
[16] "fb" is an shortcut for "facebook"

### Alchemy API concept tagging

The concept tagging API call from Alchemy[17] is capable to automatically find high level concepts(tags) which are related to the text. The concept tagging API call works so far(September 2013) only for English language. The response of the call, delivers a set of concepts, the concept relevance within the document and a reference for each concept to link data services such as Dbpedia and Freebase. The XML response has the following structure [18]:

```
<results>
    <status>REQUEST_STATUS</status>
    <url>DOCUMENT_URL</url>
    <language>DOCUMENT_LANGUAGE</language>
    <text>DOCUMENT_TEXT</text>
    <concepts>
        <concept>
            <text>DETECTED_CONCEPT</text>
            <relevance>DETECTED_RELEVANCE</relevance>
            <website>WEBSITE</website>
            <geo>LATITUDE LONGITUDE</geo>
            <dbpedia>LINKED_DATA_DBPEDIA</dbpedia>
            <yago>LINKED_DATA_YAGO</yago>
            <opencyc>LINKED_DATA_OPENCYC</opencyc>
            <freebase>LINKED_DATA_FREEBASE</freebase>
            <ciaFactbook>LINKED_DATA_FACTBOOK</ciaFactbook>
            <census>LINKED_DATA_CENSUS</census>
            <geonames>LINKED_DATA_GEONAMES</geonames>
            <crunchbase>CRUNCHBASE_WEB_LINK</crunchbase>
        </concept>
    </concepts>
</results>
```

The concept tagging method found the following concepts for these sentences:

- I like football, basketball and watching TV.
  *Result: Television, Vacuum Tube*
- Michael Jackson is my favorite musician.
  *Result: Michael Jackson*
- I am going on holidays to Barcelona.
  *Result: -*

[17] http://www.alchemyapi.com/api/concept-tagging/
[18] http://www.alchemyapi.com/api/keyword/htmlc.html

- Ikr[19] smh[20] he asked fir[21] yo[22] last name so he can add u[23] on fb[24] lololol
  *Result: -*

In general, the concept tagging call delivers few high level concepts which describe the content on a higher abstraction level. Furthermore, the returned concepts are unique, and the longer the document provided for concept tagging, the more accurate the found concept are.

[19]"Ikr" means "I know right"
[20]"smh" means "somehow"
[21]"fir" is a misspelling or spelling variant of the preposition for.
[22]"yo" is being used as equivalent to "your"
[23]"u" is an shortcut for "you"
[24]"fb" is an shortcut for "facebook"

| Tag | Description | Examples |
|---|---|---|
| **N** | common noun (NN, NNS) | books someone |
| **O** | pronoun (personal/WH; not possessive; PRP, WP) | it you u meeee |
| **S** | nominal + possessive | books' someone's |
| **^** | proper noun (NNP, NNPS) | lebron usa iPad |
| **Z** | proper noun + possessive | America's |
| **L** | nominal + verbal | he's book'll iono (= I dont't know) |
| **M** | proper noun + verbal | Mark'll |
| **V** | verb incl. copula, auxiliaries (V*, MD) | might gonna ought couldn't is eats |
| **A** | adjective (J*) | good fav lil |
| **R** | adverb (R*,WRB) | 2(i.e.,too) |
| **!** | interjection | lol haha FTW yea right |
| **D** | determiner | the teh its it's |
| **P** | pre- or postposition, or subordinating conjunction | while to for 2 (i.e., to) 4(i.e.,for) |
| **&** | coordinating conjunction(CC) | and n & + BUT |
| **T** | verb particle | lol haha FTW yea right |
| **X** | existential there, predeterminers | both |
| **Y** | X + verbal | there's all's |
| **#** | hashtag (indicates topic/category for Tweet) | #acl |
| **@** | at-mention (indicates another user as a recipient of a Tweet) | @BarackObama |
| **~** | discourse marker, indications of continuation of a message across multiple Tweets | RT and : in retweet construction RT @user : hello |
| **U** | URL or email address | http://bit.ly/xyz |
| **E** | emoticon | :-) :b (:<3 o_O |
| **U** | URL or email address | http://bit.ly/xyz |
| **E** | emoticon | :-) :b (:<3 o_O |
| **$** | numeral | 2010 four 9:30 |
| **,** | punctuation | !!! .... ?!? |
| **G** | other abbreviations, foreign words, possessive endings, symbols, garbage | ily (i love you) |

**Table 2.4:** Part-of-Speech Tagset for Twitter [9].

# Chapter 3

# Experimental Setup

To address the research questions of this thesis, an empirical study was set up which explores how predictable repliers are on the social media applications Twitter and Boards.ie, and to what extent users' reply behavior is driven by topical and social factors. The approach to answer these questions was to find out what is the major difference between a user who saw a certain message and answered, and a user who saw the same message but did not answer. In the first step of this study a representative dataset was created, described in Section 3.1. To quantify the aforementioned difference a comprehensive set of features is defined, presented in Section 3.2. To explore the predictive power of each feature a significance analysis is performed, which is outlined in more detail in Section 3.3. In a prediction experiment, users are randomly chosen from the potential reply candidates of several messages and are classified as either repliers or non-repliers depending on their feature values. Furthermore, the prediction accuracy of all classified samples is calculated. Section 3.3 describes the methods used for the prediction experiment in more detail.

## 3.1 Datasets

To carry out the empirical studies in this thesis, a dataset has to be created first for Twitter and Boards.ie. The dataset generation focuses on creating samples containing $< author - candidate >$ pairs, where either a user *c (candidate)* saw a message *m* authored by user *a (author)* and replied to it, or where a user *c* saw a message *m* authored by user a and did not reply to it. In addition to the creation of samples, the dataset generation also includes gathering appropriate user information to calculate user-centric features.

To gather many $< author - candidate >$ pairs many conversations were collected from each social media application. In this thesis a conversation is defined as an interaction between at least two users, consisting of at least two messages, the original start

message and the reply message. Each conversation starts with an initial message. The author of this initial message refers to the author in the $< author - candidate >$ user pairs. This message is seen by a set of users, the potential reply candidates. Inherently, to answer to a message one has to see the message first. Out of the set of potential reply candidates two groups can be identified. The first group contains users who saw the first message and replied to it (replier group). The second group contains users who saw the message, but decided not to reply (non-replier group). The samples which contain $< author - candidate >$ pairs, in which the candidate belong to the replier group are called positive samples, while samples which contain candidates of the non-replier group are called negative samples. Figure 3.1 illustrates this issue. The exact way the $< author - candidate >$ pairs are created for Twitter and Boards.ie, is explained in Section 3.1.1 for Twitter and 3.1.2 for Boards.ie.

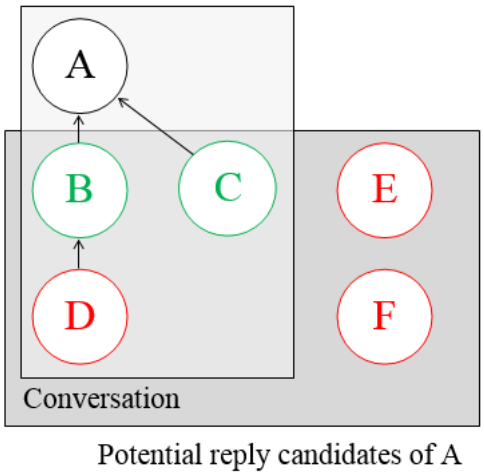


**Figure 3.1:** This figure illustrates a conversation consisting of two branches having four participants. User A is the initiator of the conversation. There is a group of potential reply candidates, who see the message of A (B, C, D, E and F). The users B and C see the message of the author and reply to him/her and, thus, belong to the replier group. User D, who participates in the conversation, is defined not to be a replier of A, because he/she replies to B and not directly to A. But D is likely to see the message of A and, therefore, D belongs to the group of non-repliers. Users E and F see the author's message but do not answer and, therefore, are members of the group of non-repliers.

Estimating who will see a message requires approximating unobservable variables, e.g. the time a user spends reading messages, the user attention to the author and so on. For this issue, a simplification will be used in this thesis which differs for Twitter and Boards.ie. The exact way the candidates are picked for the negative samples is described in Section 3.1.1 for Twitter and 3.1.2 for Boards.ie.

Another simplification was to take only the first and second message within a conversation to gather $< author, candidate >$ pairs. The main reason for this is that the first replier in the conversation has a clear reference to the author of the first message. In case of the second replier this reference is less clear, because it is not possible to distinguish whether the reply action was influenced by the first replier or only by the author of the first message. Therefore, only the first replier is taken into account. For each user occurring in the positive and negative samples, further information has to be gathered to calculate all feature values for each sample.

As a result of the dataset generation, we have a set of positive and negative samples containing $< author, candidate >$ pairs and user information necessary for calculating the features for each sample. In the following sections, the dataset creation for Twitter and Boards.ie is explained in more detail.

### 3.1.1 Twitter Dataset

#### Crawling Conversations on Twitter

Crawling conversations in Twitter is a non-trivial task. The Twitter API does not offer a request to directly search for reply messages or for conversations. Nevertheless, there are methods which make the reconstruction of conversations possible.

A conversation in Twitter can be regarded as a tree structure with a single top level node. Starting with this single message, it is possible that more than one user replies to this message and therefore a conversation can have several branches at each level. In Figure 3.2 a schema of a Twitter conversation is illustrated.

Two ways of finding messages which belong to a conversation are the following:

- *in_reply_to_status_id*: Each Tweet returned by the Twitter API includes the *in_reply_to_status_id* attribute, which refers to the Tweet ID of the replied message. If the crawled Tweet is not a reply, the *in_reply_to_status_id* is set to Null. The Twitter API does not offer a direct request for finding messages which have the *in_reply_to_status_id* attribute set.

- @username: Twitter users often use the @ sign as a marker to address other users, as in @sandy12 indicating that the message is addressed to the user with the username sandy12. The @ sign indicates in around 91% of the cases that the message is addressed to another user [11]. The Twitter REST API offers the statuses/mentions request to gather messages where a specified user was mentioned.

Furthermore, a common method to crawl English conversation on Twitter is described in more detail. This method was used in this thesis and is further mentioned

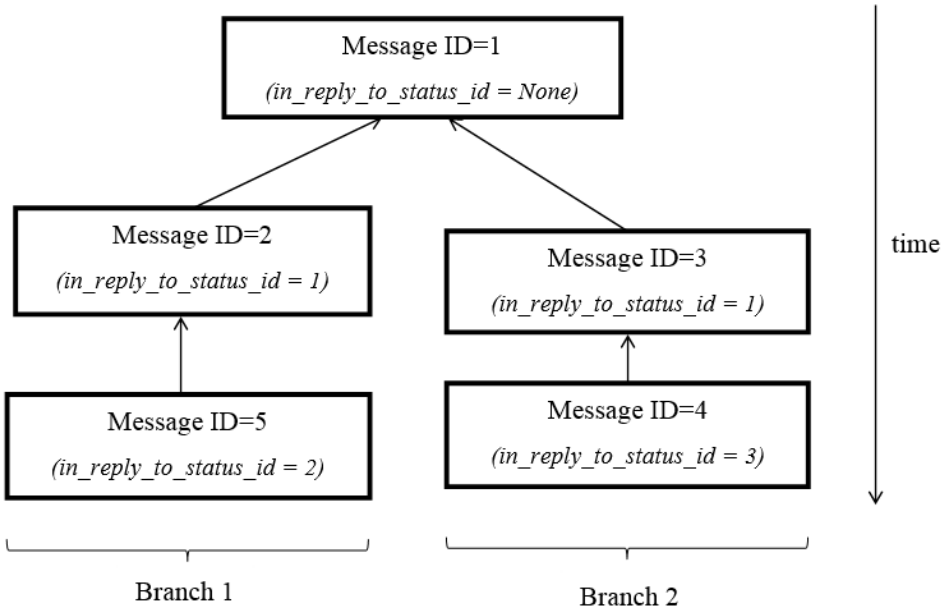


**Figure 3.2:** Schematic view of a Twitter conversation.

in [5]. To gather Twitter conversation, Twitter's public timeline[1] was crawled by using its Streaming API. After this, all English messages containing a $reply\_to\_status\_id$ - i.e., Tweets which were published in reply to another message - were filtered. Since those Tweets are part of a conversation, the conversation thread is reconstructed by recursively crawling all past messages which belong to this conversation. The tool used for crawling is outlined in Section A.2.1.

### Dataset Generation

First, there are 3,850 random conversations on November 20th, 2012 obtained by using the above mentioned crawling method.

For each conversation there is exactly one positive author-candidate pair which consists of the author of the start message of the conversation and the first user who replied to this message. Further, for each of the conversations one negative sample is created, by selecting one follower of the author of the start message who has not replied to it. One assumption of this work is that the followers of a user are those users who are likely to see a message authored by this user. To make sure that positive and negative samples are constructed in a consistent way, only positive author-candidate pairs are taken where the candidate is a follower of the author of the start message. In this way around 19% of the sample conversations are removed since users who were not following the author of

[1] https://dev.twitter.com/docs/api/1/get/statuses/publictimeline

the message replied to it. This finding confirms the finding of [21] who found that 9% of answerers are not following the askers.

Figure 3.3 illustrates the process of gathering one positive and negative samples from a Twitter conversation and shows some restrictions of our approach. One restriction, which occurs when using the crawling method described in Section 3.1.1, is that only one branch of a conversation is captured. Therefore it is possible that one replier of a message is not captured and can be falsely put in the non-replier group. Since around 92% [17] of Twitter conversations consist of two users, they also have only one branch. And in case of conversations having several branches the chance of picking the replier of the not captured branch from the followers is on average very small. Therefore this false classification can be neglected.

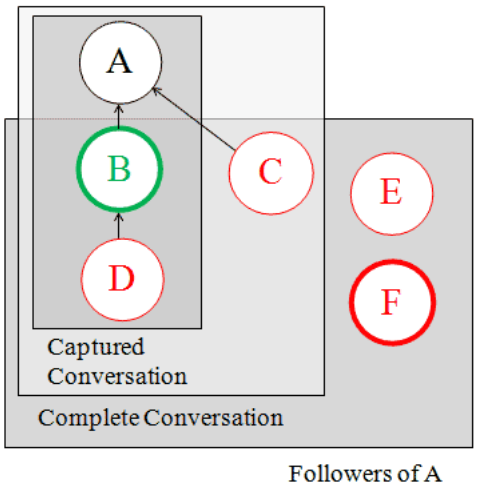


**Figure 3.3:** This figure illustrates the creation of one positive and negative sample out of a conversation, and shows some restrictions of the used conversation crawling method. Because our conversation crawler can only capture one branch of a conversation, user C would not be recognized as a replier. B belongs to the replier group because he/she was replying to A and is a follower of A. C, D, E and F are in this example followers of A and , thus, belong to the group of non-repliers. The positive sample in this conversation would be $< A, B >$. For the negative sample one user is randomly picked from the followers of A (except of B), for instance the user F. The resulting negative sample would be $< A, F >$.

I ended up having 3,215 positive and 3,215 negative samples. For all users who are part of the positive or negative samples (containing 9122 users) I further crawled their most recently published messages (up to 3,200 Tweets), their user list memberships, the user lists they created, their user profile information and their followers and followees. It is checked that there are no duplicate author/candidate pairs in the positive and negative

samples. I want to point out that this information was crawled one day after the conversations were crawled, on the 21th of November 2012. This implies that the information about user's social network, their users lists and their biography may have changed during that day. Therefore features which are based on this information may contain future information which was not available when the conversation happened.

Table 3.1 shows the basic characteristics of the Twitter dataset. The zero median value for the number of participating membership lists and the created membership lists per user indicates that many user do not use or create membership lists. Further one can see from the table that the number of followers per user have a high standard deviation since outliers with multiple millions of followers are included in the dataset.

| | **median** | **mean** | **std** |
|---|---|---|---|
| Conversation length | 3.0 | 5.3 | 12.2 |
| Tweets per user | 1,991.9 | 1,702.2 | 1,047.7 |
| List memberships per user | 0.0 | 33.2 | 456.2 |
| Created lists per user | 0.0 | 0.1 | 0.7 |
| Character length of bio information per user | 73.4 | 68.7 | 52.4 |
| Followers | 266.0 | 1,524.1 | 13,819.7 |
| Followees | 295.7 | 1,205.2 | 8,237.7 |

**Table 3.1:** Characteristics of the Twitter dataset consisting of 3,850 conversations and 12,701 users.

### 3.1.2 Boards.ie Dataset

#### Dataset Generation

The Boards.ie experiment is based on a dataset[2] which includes ten years of discussion from 1998 to 2008 within the Irish bulletin board Boards.ie. The dataset is structured based on a top-down link structure, where the top-level site document links to different top-level forums such as Arts, Sports or Tech. These forums link to sub-forums and threads, which finally link to different posts. An example of a very active sub forum is the Computer and technology forum. A Boards.ie thread equates in this thesis an conversation. Every thread(conversation) starts with an initial message and contains all posts in chronological order. New messages within a posts are added at the beginning of the thread. The dataset contains also linked list between posts, so it is possible to find out to which post another post was replying to.

[2]http://www.icwsm.org/2012/submitting/datasets/

In the experiment, the threads of the Computer and Technology forum from the year 2006 are used for extracting positive and negative samples. In this way around 2900 conversations were extracted. For each conversation, exactly one positive author-candidate pair was created, which consist of the initial message of the conversation and the first replier. To estimate the users who are likely to see the authors message, needed for the candidates in the negative samples, users are picked who wrote at least one message in the Computer and Technology forum within the last 14 days before the authors message was written. The reason why 14 days was chosen is that, the average time a user is active in the Computer and technology forum is also around 14 days. This should prevent, that the candidates of the negative samples include users who were active only long time ago before the start message of the conversation was written. In this way 1432 positive and 1432 negative samples are created. Compared to the Twitter samples, where none of the users appears twice in the positive and negative samples, there is a high overlap of similar candidates in the positive and negative samples. In the 1432 positive Samples there are 230 unique candidates and in the negative samples 935 unique candidates. This indicates an existence of users in the positive samples who are very active.

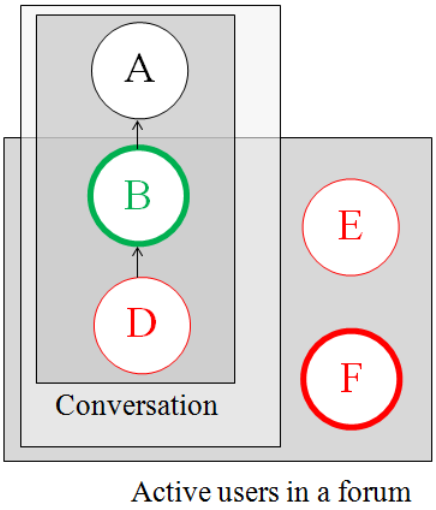


**Figure 3.4:** This figure illustrate the creation of one positive and negative sample in Boards.ie. User B in this example belongs to the replier group because he/she was the first replier of the initiator A of the conversation. The group of non-repliers consists of the users D, E and F, because they were active within the last 14 days the message of user A was written. D is not a replier of A, because it is not possible to prove that users' D reply was influenced by user B. The positive sample in this case would be $< A, B >$, and one possible solution for a negative sample could be $< A, F >$.

## 3.2 Feature Engineering

To quantify the influence of different factors (features) on the user communication behavior, the task was to find relevant features. A natural assumption when thinking of why someone replies to a user would be that users with similar interests are more likely to speak with each other. Another assumption would be that the social relation to the author of the message has a strong impact whether a person is replying or not. Users who are stronger related have probably a higher chance to speak with each other. Further, one can assume that some users who are in general more communicative, have a greater likelihood to reply, for example, in a group conference the people who are more communicative are probably more likely to reply to a person than other participants.

In this thesis features are first divided into similarity/relational and user centric features. While similarity/relational features quantify the similarity or relation between two users, the user centric features focus on the characteristic of a single user such as his/her activity in the social media application. Inspired by Chen et al. [4] who recommended conversations based on their topical relevance and the tie strength among participants, I further separate the similarity/relational features in topical and social features. These three categories are capable to refer to the above mentioned assumptions, and are defined as follows:

- **Topical features**
  Topical features capture the topical similarity between the author of a message and the potential reply candidates. To extract relevant topics from messages the following three methods are evaluated:
  - Twitter POS-Tagger (Section 2.3.2)
  - Keyword extraction using the Alchemy API (Section 2.3.3)
  - Concept tagging using the Alchemy API (Section 2.3.3 )

  When using the POS Tagger, there are only proper nouns and hashtags extracted from the text. For calculating the similarity between two lists A and B of topics the cosine similarity measure was used which is defined as follows:

$$cosine\_similarity(A, B) = \frac{A \cdot B}{||A|| * ||B||} \tag{3.1}$$

  Further, the applicability of each topic extraction method is evaluated in the significance analysis described in Section 3.3.1 for each platform.

- **Social features**
  Social features capture the strength of relation between the author and the candidate. The set of features differ strongly for Twitter and Boards.ie, because they

provide different kind of data which give a hint about the relation between users. Depending on the platform, features are calculated which refer to the past communication, the mutual reply partner overlap and also features which refer to the friends overlap between the author and the candidate for each sample. For computing the overlap between two sets of users the Jaccard similarity is used which is defined as follows:

$$Jaccard\_index(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.2}$$

- **Activity/Popularity features**
  The third category of features are the activity features. These features capture how active or communicative, and also how popular a reply candidate is. Activity features do not measure any association between the reply candidate and the author but rely solely on characteristics of the candidate. Activity features represent common confounding variables since they might be correlated with some topical and social features. Activity features represent of course not the only confounding factor. For example, external events or happenings or users' current locations might be other confounding variables. However, those factors can unfortunately not be obtained from our observational dataset. However, since the positive and negative samples are constructed randomly (with a slight bias towards active users in the case of positive samples) it can be assumed that other confounding factors are equally distributed across positive and negative samples

All feature values are normalized by firstly subtracting the mean in each feature and secondly dividing the values of each feature by its standard deviation. The normalization is defined as follows:

$$x\_norm = \frac{x - \mu}{\sigma} \tag{3.3}$$

where $\mu$ denotes the mean of the feature $X$ and $\sigma$ refer to the standard deviation of $X$. Consequently, values of each feature have zero-mean and unit-variance

### 3.2.1 Twitter Features

In the following all features used for the Twitter experiment are listed.

#### Topical Features

The following three methods are used for representing users as documents:

- First, each user is represented as an aggregation of messages which he/she recently published (up to 3,200).
- Second, each user is represented as an aggregation of the names and descriptions of the user lists he/she is a member of.
- Third, each user is represented by his/her personal description obtained from his/her user profile page.

Each topic annotation method combined with each document representation method provides a different topic vector for a user and allows computing the topical similarity between the author of a message and the potential reply candidate based on their topic vectors. Table 3.2 shows the mean number of topics which can be obtained for a user using the different types of user information and the three different topic extraction methods. Not surprisingly, Tweets allow to obtain the highest number of topics per user, followed by lists and bio information. The Twitter POS Tagger extracts in average(mean) the most number of topics from Tweets and lists, while for the bio information the keyword extraction method from Alchemy extracts slightly more topics than the others.

Using the three aforementioned methods for representing users via documents, applying three different topic extraction methods to extract topics from these documents and using cosine similarity as similarity measure, for each pair of users $< a, c >$ the following features are computed: The $TweetPostagsSimilarity$, $TweetKeywordSimilarity$ and $TweetConceptSimilarity$ describes how similar two users are, given their topics (postags, keywords, concepts) they are tweeting about. The $ListPostagsSimilarity$, $ListKeywordSimilarity$ and$ListConceptSimilarity$ describes how similar users' list memberships are, given the topics the lists are about. Finally, the $BioPostagsSimilarity$, and $BioKeywordSimilarity$ $BioConceptSimilarity$ reveals how topically similar two users are, given the topics extracted from their personal descriptions on Twitter.

These features are calculated for each of the three topic extraction methods separately. When using the term keyword in topical features, it is meant topics which are extracted using the Alchemy Keyword extraction method. Concepts in topical features refer to topics extracted with the Alchemy concept tagging method and when using the term postags, topics are meant which are extracted with the Twitter POS Tagger.

### Social Features

Social features capture the strength of the social relation between the author a and a reply candidate c. This thesis introduces the following six social features: The $NumReplyRelation$ feature describes how often the reply candidate has communicated with the author in the past. The $ReplyPartnerOverlap$ feature reveals if the author and the reply candidate tend to have similar communication partners. The $FriendsOverlap$ feature describes

| | median | mean | std |
|---|---|---|---|
| **Twitter POS Tagger** | | | |
| Tweet postags per user | 455.0 | 608.1 | 680.4 |
| List postags per user | 0.0 | 18.7 | 160.4 |
| Bio postags per user | 1.0 | 1.8 | 2.5 |
| **Alchemy Keyword Extracting** | | | |
| Tweet concepts per user | 50.0 | 67.3 | 35.5 |
| List concepts per user | 0.0 | 3.4 | 9.2 |
| Bio concepts per user | 2.0 | 2.7 | 2.7 |
| **Alchemy Concept Tagging** | | | |
| Tweet concepts per user | 8.0 | 7.4 | 4.3 |
| List concepts per user | 0.0 | 4.8 | 12.1 |
| Bio concepts per user | 1.0 | 1.5 | 1.6 |

**Table 3.2:** Number of postags/keywords/concepts per user based on the three different types of user information (Tweets, bio, lists) which are available on Twitter.

how many similar friends the author and the reply candidate have in their follower/followee network. The $isFriend$ feature is a boolean value describing if the author and candidate have a bidirectional follower/followee relation or not. The $CommonListMembership$ feature measures the overlap between the list memberships of the author and the candidate - i.e. in how many common lists they are both members. Finally, the $CandInAuthorsListfeature$ measures the overlap between the lists the author has created and the lists the candidate is member of.

For computing the overlap between the set of users or lists related with the author $a(users(a)orlists(a))$ and the set of users or lists related with the potential reply candidate $c(users(c)orlists(c))$ the Jaccard similarity coefficient is used.

**Activity Features:**

The following six activity features as follows are computed: The $TweetActivity$ feature measures the general activity level of a user on Twitter based on the number of Tweets he/she has written in the past. The $AvgTweetActivityLastWeek$ feature measures the user's average Tweet activity per day within the last week. The $ReplyActivity$ feature shows how communicative a user is given the number of reply messages the user has written in the past. The $Openness$ feature reveals how open a user is giving the number of users he/she is communicating with. The $Followers$ feature captures the popularity of a user given his/her number of followers. The $Followees$ feature indicates the number of users a user is interested in given his/her number of followees.

Table 3.3 gives an overview of all features described above.

| Feature | Description | Mathematical Description |
|---|---|---|
| ***Topical Features*** | | |
| TweetKeywordSimilarity<br>TweetConceptSimilarity<br>TweetPostagsSimilarity | Cosine similarity between the keywords/concepts/postags vectors extracted from the Tweets of candidate $c$ and author $a$. | $\frac{tweet\_concepts(a) \cdot tweet\_concepts(c)}{\|\|tweet\_concepts(a)\|\| * \|\|tweet\_concepts(c)\|\|}$ |
| BioKeywordSimilarity<br>BioConceptSimilarity<br>BioPostagsSimilarity | Cosine similarity between the keyword/concept/postags vectors extracted from the bio information of candidate $c$ and author $a$. | $\frac{profile\_concepts(a) \cdot profile\_concepts(c)}{\|\|profile\_concepts(a)\|\| * \|\|profile\_concepts(c)\|\|}$ |
| ListKeywordtSimilarity<br>ListConceptSimilarity<br>ListPostagsSimilarity | Cosine similarity between the keyword/concept/postags vectors extracted from the lists candidate $c$ and author $a$ are member of. | $\frac{list\_concepts(a) \cdot list\_concepts(c)}{\|\|list\_concepts(a)\|\| * \|\|list\_concepts(c)\|\|}$ |
| ***Social Features*** | | |
| CommonListMembership | Jaccard similarity between the list memberships of candidate $c$ and author $a$. | $\frac{\|lists(a) \cap lists(c)\|}{\|lists(a) \cup lists(c)\|}$ |
| CandInAuthorsList | Number of lists created by author $a$ in which candidate $c$ is member of. | $\frac{\|created\_lists(a) \cap created\_lists(c)\|}{\|created\_lists(a) \cup created\_lists(c)\|}$ |
| NumRepliesRelation | Number of times candidate $c$ replied to author $a$ in the past. | $replies(a, c)$ |
| ReplyPartnerOverlap | Jaccard similarity between the users candidate $c$ and author $a$ have talked to. | $\frac{\|reply\_partner(a) \cap reply\_partner(c)\|}{\|reply\_partner(a) \cup reply\_partner(c)\|}$ |
| isFriend | Is true if candidate $c$ is a follower of author $a$ and vice versa. | $isFollowing(a, c) \cap isFollowedBy(a, c)$ |
| FriendsOverlap | Jaccard similarity between candidate $c$ and author $a$ given their *friends*. | $\frac{\|friends(a) \cap friends(c)\|}{\|friends(a) \cup friends(c)\|}$ |
| ***Activity Features*** | | |
| TweetActivity | Number of Tweets published by the candidate $c$. | $num\_tweets(c)$ |
| ReplyActivity | Number of replies the candidate $c$ was publishing. | $num\_replies(c)$ |
| AvgTweetActivityLastWeek | Average Tweets per day the candidate $c$ was writing within the last week. | $avg\_tweets\_week(c)$ |
| Openness | Number of users the candidate $c$ was replying to. | $num\_replyingto(c)$ |
| Followers | Number of *followers* of the candidate $c$. | $num\_followers(c)$ |
| Followees | Number of *followees* of the candidate $c$. | $num\_followees(c)$ |

**Table 3.3:** Overview of the features used in our empirical Twitter study.

### 3.2.2 Boards.ie Features

This section gives an overview of all features used in the Boards.ie experiments.

#### Topical Features

For Boards.ie each user is represented as a document containing all posts written by a user in the forum. Each topic annotation method applied on this document representation, gives a topic vector for a user and allows computing the topical similarity between the author of a message and the potential reply candidate. For calculating the similarity between two topic vectors, the cosine similarity is used.

In this way the following features are calculated: The *PostKeywordSimilarity* uses the Alchemy Keyword Extraction method to extract topics from a users written messages. The *PostConceptSimilarity* is based on Alchemy's Concept tagging method, and the *PostPostagsFeature* uses the Twitter POS Tagger for topic extraction. In case of the Twitter POS Tagger, only proper nouns are extracted from a users messages.

Table 3.4 illustrates, that the Twitter POS Tagger extracts the most topics(proper nouns) from the aggregation from a users message, followed by the Alchemy keyword extraction. The Alchemy concept tagging delivers only a few topics, in comparison to the other methods.

| | median | mean | std |
|---|---|---|---|
| **Twitter POS Tagger** | | | |
| Posts postags per user | 327.0 | 303.6 | 211.2 |
| **Alchemy Keyword Extracting** | | | |
| Posts keywords per user | 50.0 | 43.5 | 14.0 |
| **Alchemy Concept Tagging** | | | |
| Posts concepts per user | 5.0 | 4.9 | 2.8 |

**Table 3.4:** Number of postags/keywords/concepts per user extracted from the aggregation of a users' written messages in Boards.ie.

#### Social Features

The following two social features are introduced in this thesis:

- The *NumReplyRelation* feature describes how often the candidate has communicated with the author in the past.
- The *ReplyPartnerOverlap* feature, reveals if the author and candidate tend to have similar communication partners. For instance, user $c$ is a communication partner of $a$ when either $a$ replied to $c$ or $c$ replied to $a$ in the past.

### Activity Features

The following four activity features are calculated for the Boards.ie experiment:

- The *PostActivity* feature measures the number of all post written by an user.
- The *ReplyActivity* feature measures the number of replies written by a user.
- The *ReplyActivityLastWeek* feature measures a users post activity within the last week.
- The *Openness* feature reveals how open a user is giving the number of users he/she is communicating with.

In contrast to Twitter, all posts in Boards.ie except of the starting message of a thread are reply messages therefore the PostActivity includes also all reply messages. Table 3.5 gives an overview of all Boards.ie features.

| Feature | Description | Mathematical Description |
|---|---|---|
| ***Topical Features*** | | |
| PostKeywordSimilarity<br>PostConceptSimilarity<br>PostPostagsSimilarity | Cosine similarity between the keywords/concepts/-postags vectors extracted from all posts of candidate $c$ and author $a$. | $\frac{posts\_concepts(a) \cdot posts\_concepts(c)}{\|\|posts\_concepts(a)\|\| * \|\|posts\_concepts(c)\|\|}$ |
| ***Social Features*** | | |
| NumRepliesRelation | Number of times candidate $c$ replied to author $a$ in the past. | $replies(a, c)$ |
| ReplyPartnerOverlap | Jaccard similarity between the users candidate $c$ and author $a$ have talked to. | $\frac{\|reply\_partner(a) \cap reply\_partner(c)\|}{\|reply\_partner(a) \cup reply\_partner(c)\|}$ |
| ***Activity Features*** | | |
| PostActivity | Number of posts published by the candidate $c$. | $num\_posts(c)$ |
| ReplyActivity | Number of replies the candidate $c$ was publishing. | $num\_replies(c)$ |
| PostActivityLast2Week | Number of replies the candidate $c$ was writing within the last two weeks. | $posts\_twoweeks(c)$ |
| Openness | Number of users the candidate $c$ was replying to. | $num\_replyingto(c)$ |

**Table 3.5:** Overview of the features used in the empirical Boards.ie study.

## 3.3 Methodology

In this section the methods used for the significance analysis of the feature set and the prediction experiment are described in more detail.

### 3.3.1 Significance Analysis

To answer the first research question, which is focusing on whether the communication behavior of users is more social or topical driven, a significance analysis is performed. The significance analysis uses two different methods to calculate the impact of each feature on predicting potential repliers, which are statistical hypothesis tests and a regression analysis.

### 3.3.2 Statistical Hypothesis Tests

One of the main task of a statistical hypothesis test is to find out whether the difference in two distributions occurs randomly or not. In context to our significance analysis, this refers to the question whether the values of a certain feature have a different distribution in the positive and in the negative class or an equal one. In case of a difference in the distributions the feature is likely to have an impact on the reply prediction.

Statistical hypothesis tests provide information about the statistical significance of a feature, but not about the effect size. The effect size measures the magnitude of the difference between two distributions. Statistical significance exposes certainty that the difference among the feature distribution in the positive and negative samples does not appear randomly.

The result of a statistical hypothesis test is a probability (p-value), which states whether a certain null hypothesis can be rejected or not. The null hypothesis states that the difference between two distributions occurs by random. A p-value smaller than 0.05 indicates that there is a 5% probability that the null hypothesis is true, and this gives enough confidence that there exists a difference in both distributions. One important property of statistical hypothesis tests [16] is that a small difference in the distributions can be highly significant if the sample size is large enough. The following statistical hypothesis tests are used in this thesis:

- The Wilcoxon rank sum test for all numerical features. The reason for choosing the Wilcoxon rank sum test is that the numerical features calculated in this thesis are not supposed to be normally distributed. Compared to the t-test, the Wilcoxon rank sum does not assume that the features are normally distributed.

- The Chi-Square test is used for categorical features. A Chi-Square test can be used to analyze whether two distributions of categorical features differ from one another.

As a result of the statistical hypothesis test one can say that a feature is statistically significant for the replier and non-replier class, but it is unclear how suitable the feature is for predicting the class.

### 3.3.3 Regression Analysis

Since the statistical tests compute the significance for each individual feature without taking the combination of features into account, a logistic regression model was further used. A logistic regression model consists of a binary dependent variable, indicating in our case for each author-candidate pair whether the candidate replied to the author or not, and a combination of several numerical and categorical independent variables, referring to the topical, social and activity features in this work. To estimate the coefficients for each feature the logistic regression model is trained using the positive and negative samples from the dataset. Training a logistic regression model, refers to the task of finding the coefficients which are maximizing the likelihood to observe the training data. For this task, an implementation of a logistic regression model in the statistic program R is used. After training the logistic regression model one gets for each feature a coefficient and a p-value.

The p-value returned by the logistic regression model, provides information about the statistical significance of a feature in the model. A p-value smaller than 0.05 gives us enough evidence that a feature improves the model fit on the training data and therefore has an impact on the prediction of repliers. To gain further insights into the usefulness of individual features, the coefficients of statistical significant features can be interpreted. The coefficients returned from a logistic regression model are log-odds ratios and tell us how the log-odds of a ”success” (in our case a reply) changes with a one-unit change in the independent variable. Because all numerical features are normalized, the significant features can be ranked by its estimated coefficient. This ranking can be used to evaluate the research question RQ1. Further, the sign of the coefficients tells us whether the logodds of the model increase or decreases with the feature value. A positive sign states the probability that a user replies increases per one-unit change in the feature, while a negative sign states a decrease in the probability per one-unit change.

Before estimating the coefficients for the logistic regression model, one has deal with the collinearity issue. When the model includes features which are strongly correlated (so called collinear features), the solver to estimate the coefficients can run into numerical problems. This can lead to very unstable results in the coefficients, which is indicated

by a very high standard errors of the coefficient and even in the significance loss of features. Therefore, the correlation among all feature pairs is calculated before estimating the coefficients of the logistic regression model. For calculating the correlation the Pearson correlation coefficient is used. The Pearson correlation coefficient ranges from -1 to 1, where 0 means that there is no correlation between two features and 1 denotes that the features are the same. For our analysis I found an indication of the collinearity problems, when features which have a higher correlation coefficient than 0.75 are included into the model. There are several ways to deal with strongly correlating features. In this thesis there is always one of the highly correlating features removed.

In addition to the significance analysis of each feature in the logistic regression model, this work wants to find out which of the three topical extraction methods, either the Twitter POS tagger, the Alchemy keyword extraction or the Alchemy concept tagging, deliver the best model fit. Therefore three different logistic regression models are set up, containing all activity and social features and topical features of a certain extraction method. For each model, the R squared value was calculated to measure the model fit on the training data. The model with the highest r squared value was evaluated to use the best topic extraction method. For calculating the R squared value, Nagelkerkes pseudo $R^2$ is used which is defined as follows:

$$nagelkerkes\_pseudoR^2 = \frac{1 - \frac{L(M_{intercept})^{2/N}}{L(M_{full})}}{1 - L(M_{intercept})^{2/N}} \tag{3.4}$$

where $N$ denotes the number of samples, $L(M_{full})$ refers to the likelihood to obtain the training data when using all features and $L(M_{intercept})$ without using any feature in the logistic regression model. Nagelkerke pseudo R squared measure ranges from 0 to 1, where 1 denotes a perfect fit to the observed data and 0 the model does not fit at all. The evaluation of the best topic extraction method is done, before analyzing the significance of each single feature in the logistic regression model, and there are only these topical features used in the further analysis which were extracted with the best evaluated topic extraction method.

### 3.3.4 Prediction Experiment

In addition to looking into the utility of individual features, the predictive power of the whole model in order to answer the second research questions (To what extent are repliers on Twitter predictable?) was assessed. Therefore a 10-fold cross validation was conducted to train and test the logistic regression model on the dataset. Since the dataset is balanced, i.e. it contains an equal number of positive and negative samples, a random guesser baseline would lead to a performance of 50%. As evaluation measures the Precision, Recall and the F1-score which is the harmonic mean of Precision and

Recall are used. The precision, recall and F1-score are defined as follows:

$$precision = 2 * \frac{truepositives}{truepositives + falsepositives} \tag{3.5}$$

$$recall = 2 * \frac{truepositives}{truepositives + falsenegatives} \tag{3.6}$$

$$F1Score = 2 * \frac{precision * recall}{precision + recall} \tag{3.7}$$

Next to the Precision, Recall and F1-Score the model fit is calculated using Nagelkerke pseudo R squared measure for the following feature settings: all features, only topical, only social, only activity features. This gives additional insight into the contribution of each feature setting to the model fit on the training data.

# Chapter 4

# Experiments

## 4.1 Twitter Experiment

### 4.1.1 Results of the Significance Analysis

In this section the results of the significance analysis for Twitter are presented.

#### Statistical hypothesis tests

Figures 4.1, 4.2 and 4.3 show boxplots of all topical, social and activity features. The boxplots illustrate the distribution for each feature in the replier and non-replier class. The more the class-specific feature distributions differ, the higher the ability of these features to discriminate the two classes. One can see that the *ListKeywordSimilarity*, *BioKeywordSimilarity*,*ListConceptSimilarity*, *BioConceptSimilarity*, *ListPostagsSimilarity*, *BioPostagsSimilarity*, *CommonListMembership* and the *CandInAuthorsList* feature distributions, contain mainly zero values in their distribution. This indicates that these features do not have an impact in the prediction of repliers. For instance, in the case of the *CommonListMembership* feature, there exist many users who do not participate in any or just a few membership lists and, therefore, the overlap of the authors and candidates lists is in many cases zero.

For other features like the *TweetConceptOverlap*, *TweetKeywordOverlap*, *TweetPostagsOverlap*, *NumRepliesRelation*, *FriendsOverlap*, *ReplyPartnerOverlap*, *TweetActivity*, *ReplyActivity*, *AvgTweetActivityLastWeek*, *Friends*, as illustrated in figure 4.1, 4.2 and 4.3, the distributions for both classes differ stronger and, thus, these features seem to have a higher discriminative power. In case of the *Followers* feature, one can see that the distribution of both classes is almost equal and therefore this feature seems to have no impact.

The results from the Wilcoxon rank sum test and the Chi-Squared test show that all

| Feature | p-Values | Significance |
|---|---|---|
| **Wilcoxon Rank Sum Test (numerical features)** | | |
| TweetKeywordSimilarity | 8.0375e-116 | *** |
| ListKeywordSimilarity | 9.1280e-08 | *** |
| BioKeywordSimilarity | 5.0748e-06 | *** |
| TweetConceptSimilarity | 4.873e-112 | *** |
| ListConceptSimilarity | 3.882e-10 | *** |
| BioConceptSimilarity | 0.4008 | |
| TweetPostagsSimilarity | 2.4377e-149 | *** |
| ListPostagsSimilarity | 2.3052e-7 | *** |
| BioPostagsSimilarity | 4.0863e-08 | *** |
| CommonListMembership | 1.904e-17 | *** |
| CandInAuthorsList | 2.740e-08 | *** |
| NumRepliesRelation | 0.00e+00 | *** |
| ReplyPartnerOverlap | 3.644e-261 | *** |
| FriendsOverlap | 3.641e-120 | *** |
| TweetActivity | 3.418e-98 | *** |
| ReplyActivity | 2.948e-206 | *** |
| AvgTweetActivityLastWeek | 8.255e-238 | *** |
| Openness | 2.198731e-93 | *** |
| Followees | 1.640e-36 | *** |
| Followers | 0.151 | |
| **Chi-Squared Test (categorical features)** | | |
| isFriend | 2.2e-16 | *** |

**Table 4.1:** The results from the statistical hypothesis tests show that almost all features are statistically significant. Only the *BioConceptSimilarity* and the *Followers* have almost equal distributions in the replier and non replier class, thus, they are not statistically significant.

features except of *Followers* and *BioConceptSimilarity* are statistically significant (see Table 4.1). This indicates that these two features have an almost equal distribution in the replier and non-replier, thus, the *Followers* and *BioConceptFeature* do not have any impact in the reply prediction. One potential explanation why the *BioConceptSimilarity* seems to be irrelevant is that the bio information of users tends to be short with a mean length of 75 characters per user and that around 14% of the users do not provide any bio information. In a previous work Wagner et al. [30] found that the users' bio information is almost as useful as Tweets for predicting users' expertise. However, one needs to note that the dataset used in [30] was biased towards active expert users who had a high Wefollow7 rank, while the dataset in this work consist of average users who use Twitter for a conversational purpose. The number of followers seems to be unrelated with users' reply behavior which indicates that users' popularity does not impact their probability

of replying. The reason why many features such as the *CommonListMembership* or *BioPostagsSimilarity* are significant in the statistical hypothesis test, although it contains many zero values in their distributions, lies in the nature of statistical hypothesis tests. Statistical Hypothesis test have the property, that a small difference in two distributions can become significant with an increasing sample size.

To summarize the results of the statistical hypothesis test, one can say that when analyzing each feature independently, the *Followers* and *BioConceptFeature* are not useful at all in predicting repliers. Features such as *ListKeywordSimilarity*, *BioKeywordSimilarity*, *ListConceptSimilarity*, *ListPostagsSimilarity*, *BioPostagsSimilarity*, *CommonListMembership* and the *CandInAuthorsList* which are significant using the WilcoxonRankSum test but have obviously an almost similar distribution in the replier and non-replier class (see the boxplots in figure 4.1, 4.2 and 4.3), are likely to have only a very low predictive power. On the other hand, features like the *TweetConceptOverlap*, *TweetKeywordOverlap*, *TweetPostagsOverlap*, *NumRepliesRelation*, *FriendsOverlap*, *ReplyPartnerOverlap*, *TweetActivity*, *ReplyActivity*, *AvgTweetActivityLastWeek*, *Friends* and *isFriend* which have different distributions in both classes, seem to have a higher predictive power.

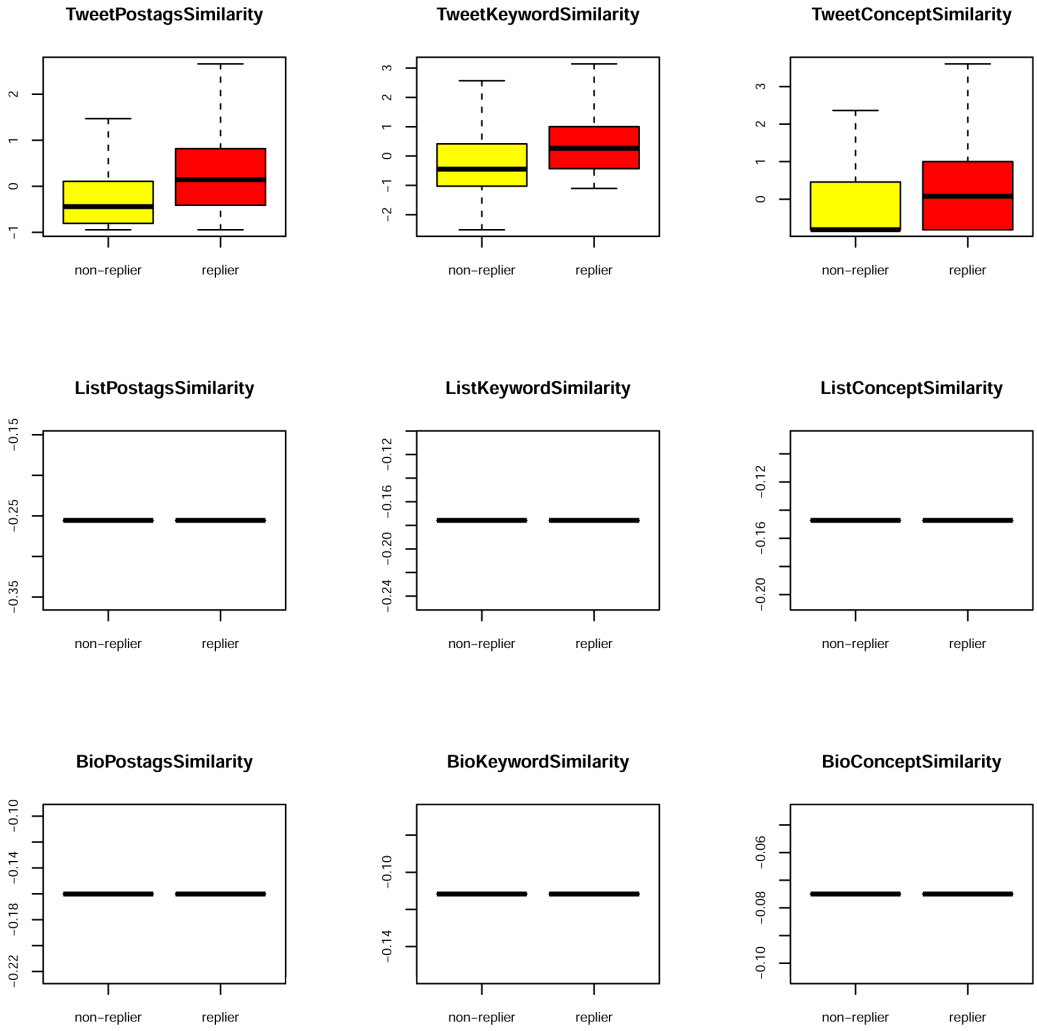


**Figure 4.1:** This figure shows the boxplots of all topical features. For each feature, the distribution in the replier and non-replier class is illustrated. The more the distributions in both classes differ the higher is the discriminative power of the feature. The *TweetPostagsSimilarity*, *TweetKeywordSimilarity* and the *TweetConceptSimilarity* differ in their replier and non-replier distributions and seem to have a higher discriminative power. For the other six features it seems at first sight that there is almost no topical overlap due to few information available.

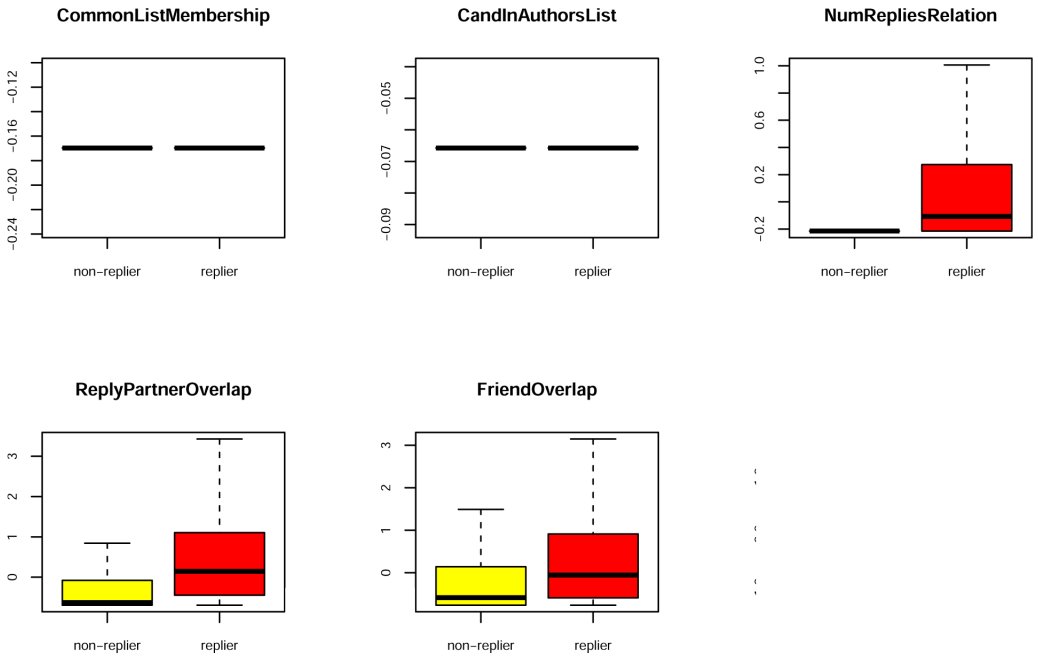


**Figure 4.2:** This figure shows the boxplots of all social features. For each feature, the distribution in the replier and non-replier class is illustrated. The *NumRepliesRelation*, *ReplyPartnerOverlap* and the *FriendOverlap* differ in their replier and non-replier distributions and seem to have a higher discriminative power. For the *CommonListMembership*, and the *CandInAuthorsList* the distributions are almost equal and therefore seems to be not relevant in the prediction of repliers.

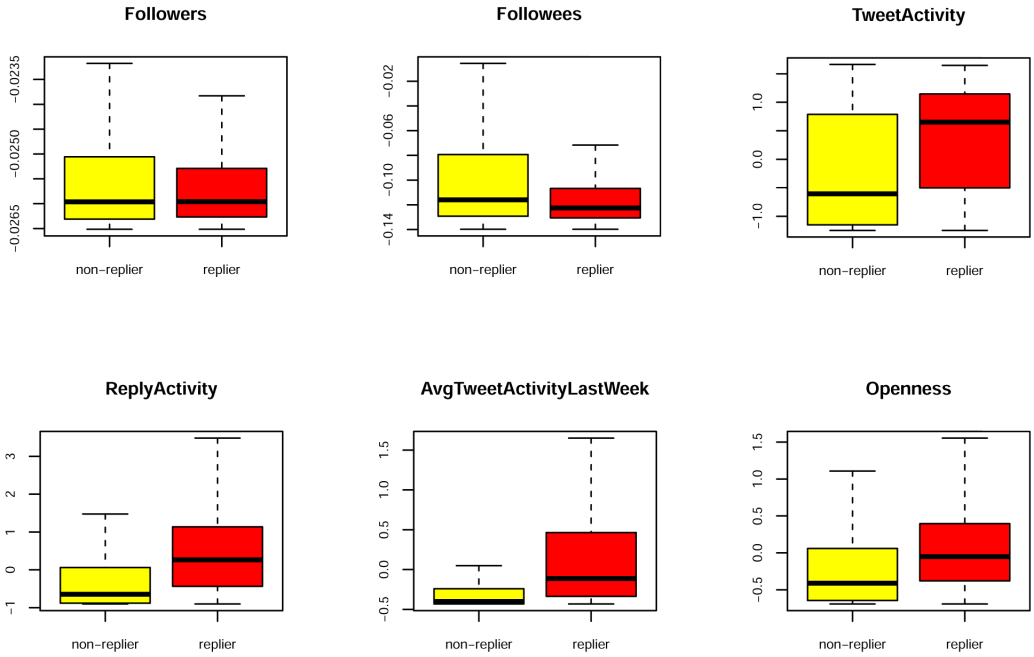


**Figure 4.3:** This figure shows the boxplots of all activity features. In case of Followers feature one can see that both distributions are almost equal. The other five features seem to have a different distribution in the replier and non-replier class.

### Regression Analysis

The first step in the regression analysis is to remove the colinearity among all features. After calculating the correlation matrix illustrated in Figure 4.4, one can see that the following features have a higher Pearson correlation coefficient than 0.75:

- *ReplyPartnerOverlap* and *FriendsOverlap* (coef = 0.80).
- *ReplyActivity* and *TweetActivity2Week* (coef = 0.82).

For the *ReplyPartnerOverlap* and *FriendsOverlap*, I decided to neglect the *FriendsOverlap* Feature because the *FriendsOverlap* is based on the followers and followees information which was crawled two days after the conversation appeared. All features, which rely on other information than a users' Tweets (e.g.,follower, followee or list membership information) may contain future information since this information was crawled two days after the conversations happened, and in theory the social network as well as the list memberships may have changed within those two days. For the *ReplyActivity* and *TweetActivity*, I decided to keep the *ReplyActivity*, because of the assumption that this feature has slightly more power to predict repliers than the *TweetActivity*.

After removing the collinearity among features, it is evaluated which of the three topical extraction methods has the best model fit. In table 4.2 one can see that topical features which are using the Twitter POS-Tagger for extracting topics, deliver a slightly better r squared value than the other methods. Therefore the POS-tagging method was evaluated to be the best of the three topical extraction methods. Further, this thesis will only use the *TweetPostagsSimilarity*, *BioPostagsSimilarity* and *ListPostagsSimilarity* in the regression analysis and in the prediction experiment.

| Topic Extraction Method | Model Fit (Pseudo R squared) |
|---|---:|
| Twitter POS Tagger | 0.402 |
| Alchemy Concept Tagging | 0.399 |
| Alchemy Keyword Extraction | 0.400 |

**Table 4.2:** The table shows for each topic extraction method the corresponding pseudo r squared value. One can see that there is only a very slight difference in the model fit, when using different topic extraction methods. The Twitter POS Tagger has a slightly higher model fit than the others therefore this method is evaluated to be the best.

Table4.3 shows the regression coefficients of each feature and their significance level. All features are normalized, so it is possible to rank their influence using their coefficients. One can see from Table4.3 that the activity features *AvgTweetActivityLastWeek* and *ReplyActivity* are significant and have a positive coefficient. This demonstrates that

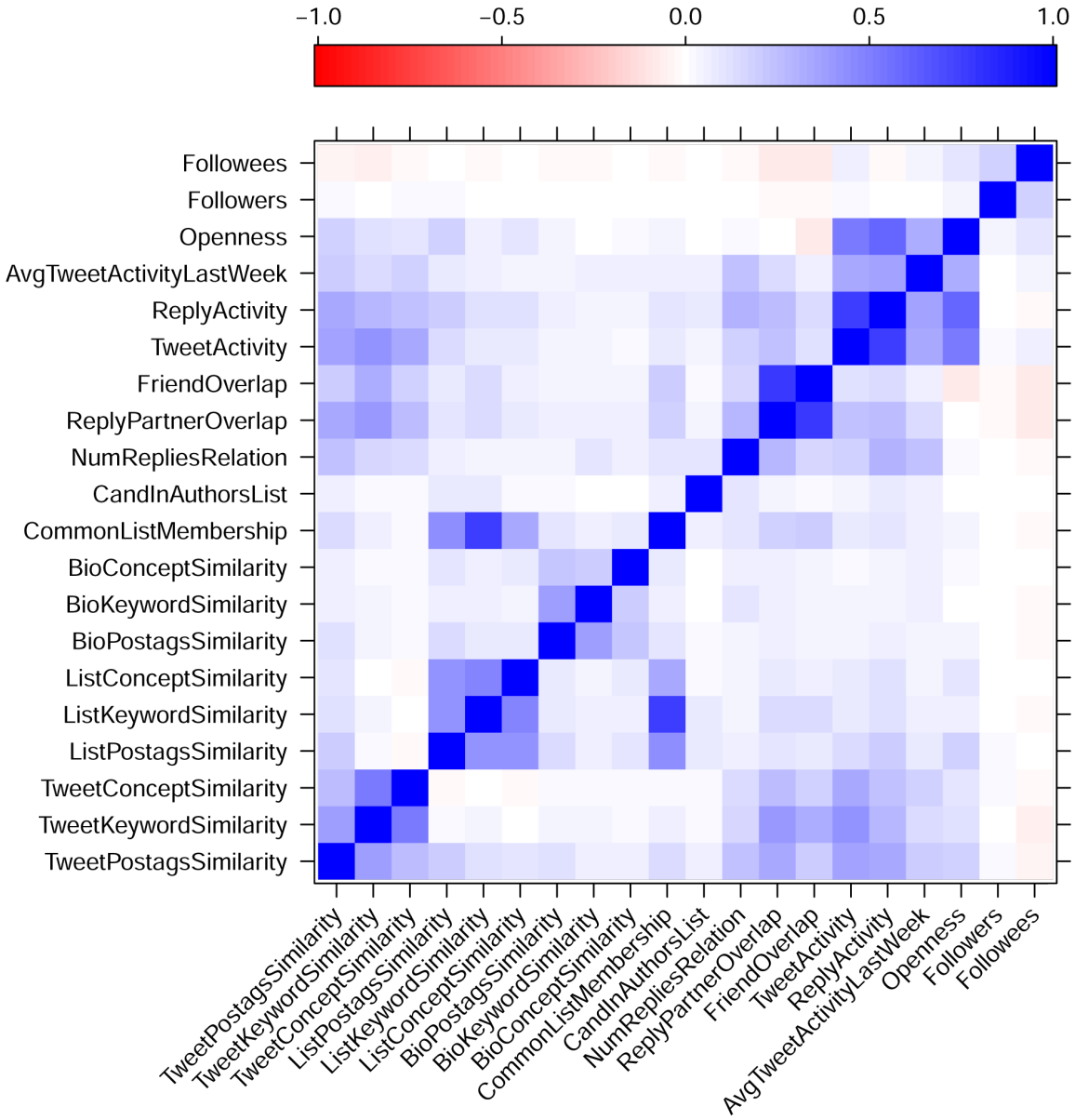


**Figure 4.4:** Pearson correlation matrix of all Twitter features. One can see from this figure that the *ReplyPartnerOverlap* and *FriendsOverlap* and the *ReplyActivity* and T*weetActivity* are strongly correlated.

the activity level of a user is indeed a significant factor, which influences if a user will reply to a message or not. Not surprisingly, active users are more likely to reply than non active users. The features which are related with the popularity and social status of a user (*Openess* and *Followers*) are not significant which means, that the users' reply behavior is not influenced by how open they are or by how many users they follow.

| | Coefficient | Significance |
|---|---|---|
| (Intercept) | 0.0402 | |
| TweetPostagsSimilarity | 0.1472 | *** |
| BioPostagsSimilarity | 0.0710 | * |
| ListPostagsSimilarity | -0.0575 | |
| NumRepliesRelation | 2.6073 | *** |
| ReplyPartnerOverlap | 0.2638 | *** |
| CommonListMembership | 0.0281 | |
| CandInAuthorsList | 0.0727 | |
| isFriend | 0.3962 | *** |
| ReplyActivity | 0.3418 | *** |
| AvgTweetActivityLastWeek | 0.3505 | *** |
| Openness | 0.0726 | |
| Followers | 0.6063 | |
| Followees | -1.9698 | *** |

**Table 4.3:** Results from the logistic regression model using topical, social and activity features as independent variables and reply or not as binary dependent variable. One can see that the topical features *TweetPostagsSimilarity* and *BioPostagsSimilarity* are significant but having only a small coefficient. The social features NumRepliesRelation, *ReplyPartnerOverlap* and *isFriend* are significant and have much higher coefficients than the topical features especially the *NumRepliesRelation*. The activity features *ReplyActivity*, *AvgTweetActivityLastWeek* and the *Followees* feature are also significant.

In addition to the activity features, the following social features have a significant positive coefficient - i.e., they help predicting repliers beyond the effects of activity features: *NumRepliesRelation*, *isFriend* and *ReplyPartnerOverlap*. This shows that previous communication relations as well as bidirectional friendship relations are very important for predicting who will reply to a message of a certain user. Friends of the author of the message who have communicated with each other before are more likely to reply than others. The only significantly negative feature is the *Followees* feature. This indicates that the more users a user is following, the less likely he/she replies to their messages, as also shown in Figure4.3. Intuitively this makes sense as it can be assumed that every user has a maximum number of Tweets to which he/she will reply e.g. per hour. The more people a user is following, the more new Tweets will show up in his/her

timeline. That means the users' reply probability is spread across more Tweets and is therefore lower for each individual Tweet.

Finally, the logistic regression model shows that topical features like the *TweetConceptSimilarity* and the *BioConceptSimilarity* are also significantly positively correlated with users' reply probability. This indicates that there is a slight tendency that users who are interested into similar topics are more likely to reply to each other. However, one needs to note that the coefficients of the significant topical features are much smaller than the coefficients of the significant social features. This indicates that users' reply behavior on Twitter is more influenced by social factors than by topical factors.

### 4.1.2 Results of the Prediction Experiment

To answer the research question RQ2, i.e. to what extent repliers are predictable in online conversations, in this case Twitter conversations, a logistic regression model was trained using the same features as in the aforementioned logistic regression model.

Our results in Table 4.4 show that when using all three types of features an average F1-score of 0.76 can be achieved while a naive baseline (random guesser) would achieve 0.5 since our dataset is balanced. The confusion matrix in Table 4.5 shows that the model classified more users who replied as non-repliers than users who did not reply as repliers. Interestingly, using social features alone was almost as good as using a combination of all features (F1=0.74). This indicates that social features contribute most to the performance of the classification model. Also, activity features alone performed very well (F1=0.70) as shown in Table 4.4. This confirms our hypothesis that the activity level of a user is a common confounding variable when analyzing the factors that influence users' reply behavior.

Finally, Table 4.4 shows that the performance is worst when using topical features alone (F1=0.63). Also Table 4.6 indicates that a logistic regression model using only topical features as independent variables is worst in explaining the variability in the training dataset, while a combination of all features is best, followed by using social features alone.

Our results clearly demonstrate that conversations on Twitter are not driven by topics but by social relations. Further our work shows that in addition to social relations users' activity level plays an important role since more active users are also more likely to reply (i.e., have a higher prior probability of replying). Researchers need to consider activity information since they may function as confounding variables when neglected. Including activity features into our models allows us to conclude that social features help predicting repliers above and beyond the effects of activity features.

| | Precision | Recall | F-Score |
|---|---|---|---|
| **All features** | | | |
| non replier class | 0.73 | 0.80 | 0.77 |
| replier class | 0.79 | 0.71 | 0.75 |
| average | 0.76 | 0.75 | 0.75 |
| **Topical features** | | | |
| non replier class | 0.61 | 0.67 | 0.64 |
| replier class | 0.63 | 0.57 | 0.6 |
| average | 0.62 | 0.62 | 0.62 |
| **Social features** | | | |
| non replier class | 0.75 | 0.85 | 0.77 |
| replier class | 0.80 | 0.63 | 0.71 |
| average | 0.75 | 0.74 | 0.74 |
| **Activity features** | | | |
| non replier class | 0.71 | 0.72 | 0.72 |
| replier class | 0.71 | 0.71 | 0.71 |
| average | 0.71 | 0.71 | 0.71 |

**Table 4.4:** Classification performance of our logistic regression model using individual feature groups and their combination.

| | predicted non replier | predicted replier |
|---|---|---|
| non replier | 2500 | 574 |
| replier | 943 | 2131 |

**Table 4.5:** Confusion matrix of the classification results using social, topical and activity features for training a logistic regression model. The columns of the confusion matrix show the predicted values and the rows show the reference values. One can see that the model classified more users who replied as non-repliers than users who did not reply as repliers.

| | all | topical | social | activity |
|---|---|---|---|---|
| $R^2$ | 0.402 | 0.105 | 0.337 | 0.246 |

**Table 4.6:** Goodness of fit of the logistic regression model measured using the Nagelkerke pseudo R squared.

## 4.2 Boards.ie Experiment

### 4.2.1 Results of the Significance Analysis

In the following sections the results from the significance analysis for Boards.ie are presented.

#### Statistical Hypothesis Tests

Figure 4.5 illustrates the boxplots of all topical, social and activity features. The boxplots give a first impression about the ability of each feature to discriminate the non-replier and replier class. Except for the *PostConceptOverlap* which contains no visible difference in both distributions, all feature seems to have an impact for the prediction of repliers. Especially the activity features show a strong difference in their distributions.

The results from the Wilcoxon rank (table 4.7) sum test indicate that all features, when looking at them individually, are statistically significant. Also the *PostConceptOverlap* feature which seems to have only a small difference in its distributions, according to figure 4.5, is statistically significant when using the Wilcoxon rank sum test.

Finally one can say at the end of the first analysis, that activity features seem to have the strongest impact in the reply behavior of users.

| Feature | p-Values | Significance |
|---|---|---|
| **Wilcoxon Rank Sum Test (numerical features)** | | |
| PostPostagsSimilarity | 9.936e-19 | *** |
| PostKeywordsSimilarity | 9.198e-13 | *** |
| PostConceptSimilarity | 4.144e-06 | *** |
| NumRepliesRelation | 2.647e-39 | *** |
| ReplyPartnerOverlap | 6.355e-14 | *** |
| PostActivity | 4.205e-98 | *** |
| ReplyActivity | 5.752e-62 | *** |
| PostActivityLast2Week | 3.680e-152 | *** |
| Openness | 4.954e-60 | *** |

**Table 4.7:** The results from the statistically hypothesis tests show that all features are statistical significant.

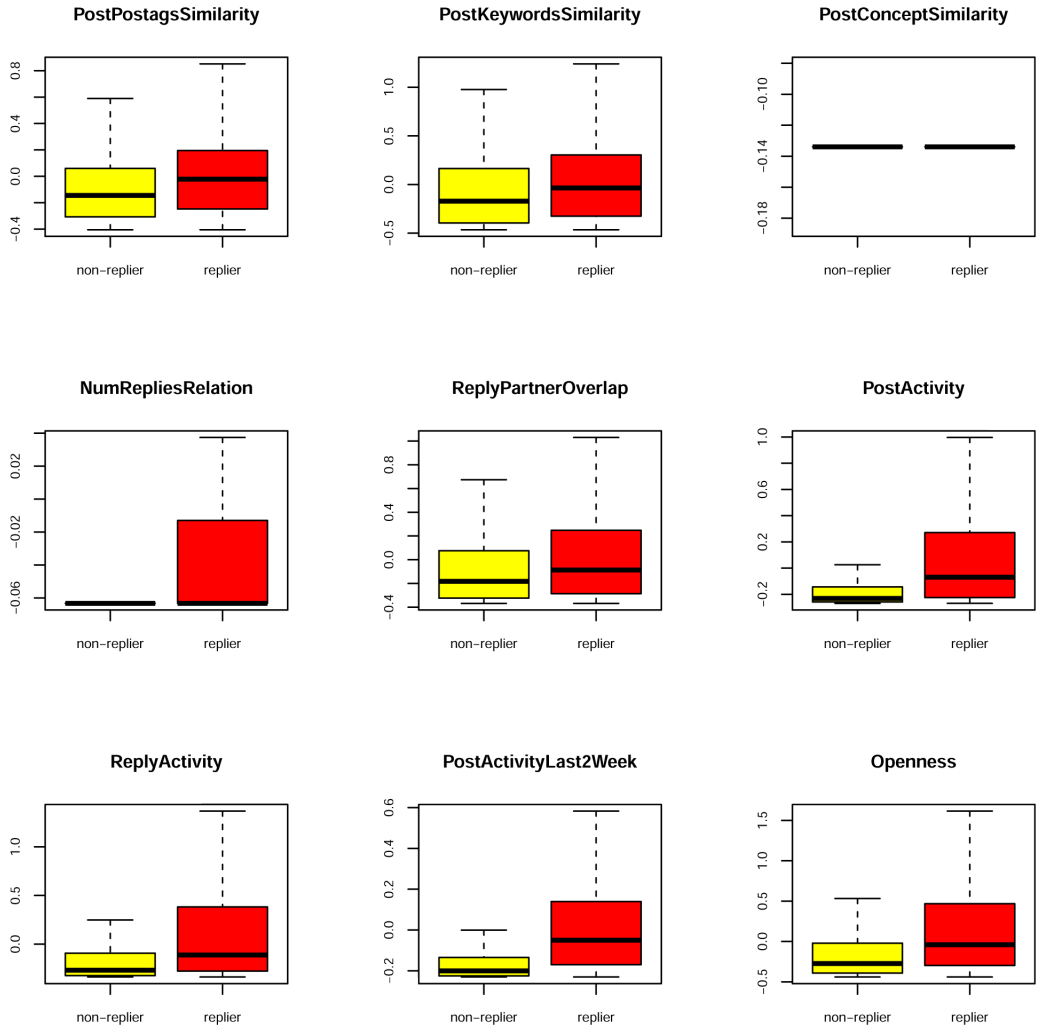


**Figure 4.5:** This figure shows all Boards.ie features and their distribution in both classes, the non-replier and replier class. One can see that the boxplots for the *PostPostagsSimiliarity* an *PostKeywordsSimiliarity* look very similar and their distributions differ in both classes. The *PostConcept-Similarity* contains many zero values, which indicates that there is only in few cases a topic overlap between author and candidate happening when using the Alchemy Concept Tagging method. The social features, *NumRepliesRelation* and the *ReplyPartnerOverlap* show a slight difference in their distributions, while all activity features show the discriminative power between the non-replier and replier class.

**Regression Analysis**

In a first step of the regression analysis, the colinearity has to be removed among all features. After calculating the correlation matrix illustrated in Figure 4.6, one can see that many features are correlating strongly (Pearson correlation coefficient >0.75). The folllowing strong correlating features could be identified:

- *PostPostagsSimilarity* and *ReplyPartnerOverlap* (coef = 0.80).
- *ReplyActivity* and *Openness* (coef = 0.94).
- *PostActivity* and *PostActivityLast2Week* (coef = 0.82).

One explanation for the strong correlation between *PostPostagsSimilarity* and *ReplyPartnerOverlap* can be that the *PostsPostagSimilariy* is proportional to the length of the topic vector extracted from the aggregation of a user's posts. The more posts a user writes, the more topics can be extracted and the higher the chance of an overlap between the topic vector of the authors and potential reply candidate. Furthermore, it seems that the more messages a user writes the more reply partners a user has. The correlation between *ReplyActivity* and *Openess* states that the more often a user replies (each message instead of the initial message is a reply in Boards.ie) the higher the number of replied users. This indicates a low preference to reply always to the same users. This observation intuitively makes sense in the context of a forum about technical topics, such as the Computer and Technology forum in Boards.ie. In these forums users mainly ask questions about technical issues and these questions are answered without preference to specific user.

The strong correlation between social and topical features indicates that both have a common bias coming from users who are more active in terms of posting more messages. To dissolve the collinearity among the above mentioned features, the *ReplPartnerOverlap*, the Openness and the *PostActivity* Feature is neglected in the further analysis.

After removing the collinearity among features, it is evaluated which of the three topical extraction methods delivers the best model fit. In table 4.8 one can see that topical features which are using the Twitter POS-Tagger for extracting topics, deliver a slightly better r squared value than the other methods. Therefore, the POS-tagging method is evaluated to be the best of the three topical extraction methods. In the further Boards.ie experiment, only the *PostPostagsSimilarity* will be used as topical feature. Table 4.9 shows the regression coefficients and the significance level for each feature. All features are normalized, so it is possible to rank their influence using their coefficients. One can see from Table4.9 that the activity feature *PostActivityLast2Week* is significant and has the highest coefficient, thus, it has the biggest impact in the prediction of replier. The *ReplyActivity*, is not significant in the logistic regression model. The reason for this is

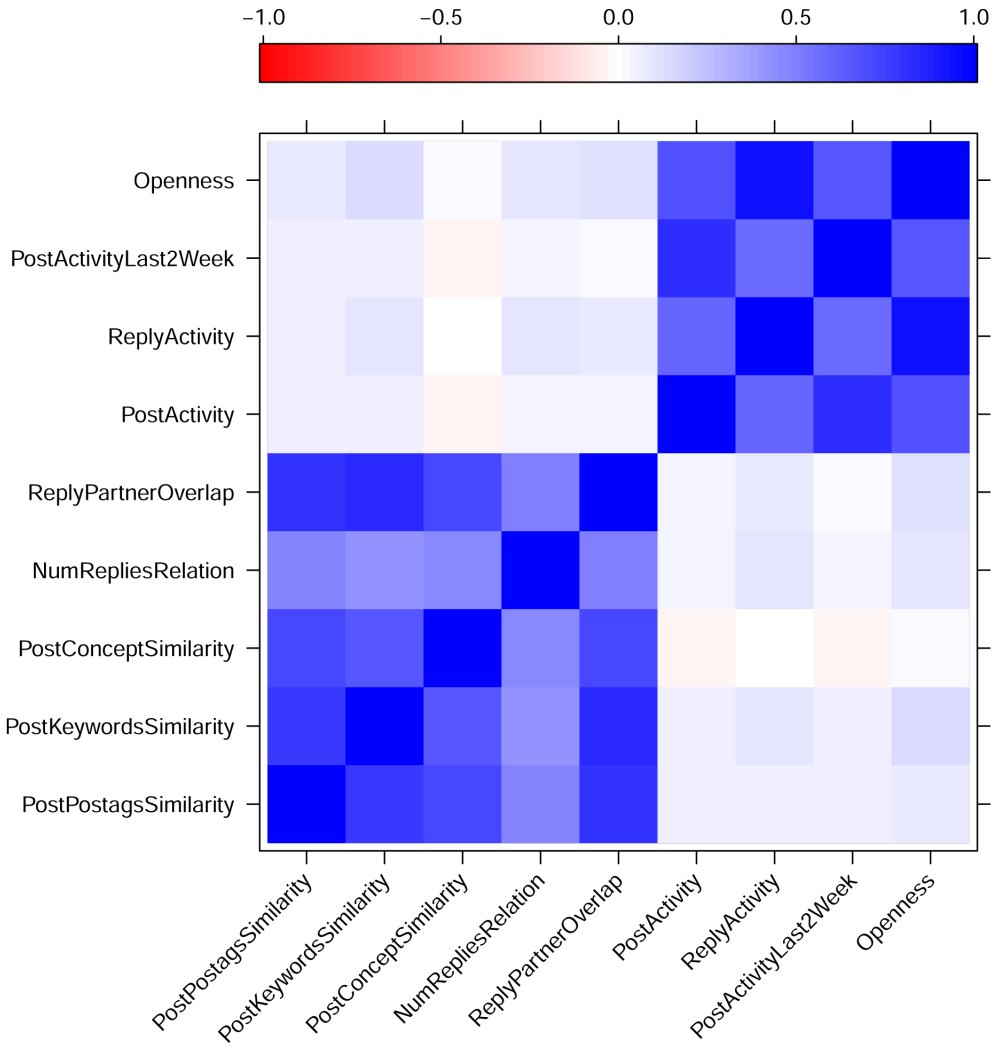


**Figure 4.6:** Pearson correlation matrix of all Boards.ie features. One can see that social and topical features are strongly correlating. Especially the features *PostPostagsSimilarity* and *ReplyPartnerOverlap*.Within the activity features there is a strong correlation between the *ReplyActivity* and *Openness*, and the *PostActivity* and *PostActivityLast2Week*.

| Topic Extraction Method | Model Fit (Pseudo R squared) |
|---|---|
| Twitter POS Tagger | 0.292 |
| Alchemy Concept Tagging | 0.288 |
| Alchemy Keyword Extraction | 0.287 |

**Table 4.8:** The table shows for each topic extraction method the corresponding pseudo r squared value. Similar to the Twitter experiment, there is only a slightly difference in the model fit, when using different topic extraction methods. The Twitter POS Tagger has a slightly higher model fit (0.292) than the others therefore this method is evaluated to be the best.

that the *ReplyActiviy* feature does not additionally improve the model fit above the highly correlating *PostActivityLast2Week* feature (Pearson correlation coefficient of 0.6). This strengthen our first analysis from the statistical hypothesis tests, that active users are more likely to reply than non active users.

| | Coefficient | Significance |
|---|---|---|
| (Intercept) | 0.504 | *** |
| PostPostagsSimilarity | 0.628 | *** |
| NumRepliesRelation | 1.932 | *** |
| PostActivityLast2Week | 5.2414 | *** |
| ReplyActivity | -0.0346 | |

**Table 4.9:** Results from the logistic regression model using topical, social and activity features as independent variables and reply or not as binary dependent variable. One can see that all features are significant, except for the *ReplyActivity*.

Further one can see in table 4.9 that the *NumRepliesRelation* feature, which refers to the number of times the $candidate$ replied to the $author$ in the past, is significant but has an almost three times lower influence in the regression model than the *PostActivityLast2Week*. However, the *NumRepliesRelation* gives an improvement in fitting the sample data. The *PostPostagsSimilarity* is also significant but has a much smaller influence than the *NumRepliesRelation* feature. Due to the high correlation among the *PostPostagsSimilarity* and *NumRepliesRelation* (Pearson correlation coefficient of 0.48), one can not make a clear statement whether the social feature, *NumRepliesRelation*, or the topical feature, *PostPostagsSimilarity*, performs better in Boards.ie in the prediction of the repliers.

As a result of the regression analysis, one can conclude that the *PostActivityLast2Week* feature, which refers to the number of written messages the last two weeks before the inital message of the conversation was written, has the highest impact in predicting repliers. Furthermore, because of the high correlation between the *ReplyActivity* and *Openness* feature, there seem to be no user preferences when replying to someone. To answer

RQ1, whether the communication behavior is more driven by social or by topical factors, it is referred to the prediction experiment in the next section.

### 4.2.2 Results of the Prediction Experiment

To answer RQ2 for Boards.ie, a logistic regression model was trained and tested using the following features: *PostPostagsSimilarity*, *NumRepliesRelation*, *ReplyActivity* and the *PostActivityLast2Week*. The classification results illustrated in table 4.10, show that when using all three types of features an average F1-score of 0.71 is achieved while a naive baseline (random guesser) would achieve 0.5 since our dataset is balanced. The confusion matrix in Table 4.11 shows that the model classified around 60% of the candidates as non-repliers and 40% are classified as repliers. Interestingly, using activity features alone was almost as good than using a combination of all features (F1=0.71). This indicates that the activity features contribute most to the performance of the classification model. The topical features alone have an F1-score of 0.55, while the social feature performs only slightly better (F1-score of 0.57). Combining all four features, brings only an improvement of 0.02 in the average F1 score than using activity features only. Table 4.12 shows similar results. The model fit of the logistic regression model when using only topical features as independent variables is worst (R squared of 0.056), followed by social features (R squared of 0.075), while a combination of all features delivers the best model fit (R squared of 0.291), followed by using activity features alone (R squared of 0.254). This shows that more active users are more likely to reply to someone than non active users. The impact of social and topical features is rather small, and can be almost neglected.

To sum up the prediction analysis, one can conclude that there is only a small tendency in the Computer and Technology Forum of Boards.ie, that users reply to someone with whom they already talked in the past. Moreover, there is also only a small tendency that users speak with people who write about similar topics. When using activity features reply partners can be predicted with an F1-score measure of 0.69. There is only a slight improvement in the F1-score of 0.02 when adding topical and social features.

| | Precision | Recall | F-Score |
|---|---|---|---|
| **All features** | | | |
| non replier class | 0.67 | 0.83 | 0.75 |
| replier class | 0.79 | 0.60 | 0.68 |
| average | 0.73 | 0.71 | 0.71 |
| **Topical features** | | | |
| non replier class | 0.56 | 0.9 | 0.69 |
| replier class | 0.74 | 0.29 | 0.41 |
| average | 0.65 | 0.59 | 0.55 |
| **Social features** | | | |
| non replier class | 0.57 | 0.67 | 0.61 |
| replier class | 0.60 | 0.50 | 0.54 |
| average | 0.58 | 0.58 | 0.57 |
| **Activity features** | | | |
| non replier class | 0.65 | 0.84 | 0.74 |
| replier class | 0.78 | 0.55 | 0.65 |
| average | 0.71 | 0.69 | 0.69 |

**Table 4.10:** Classification performance of the logistic regression model using individual feature groups and their combination.

| | predicted non replier | predicted replier |
|---|---|---|
| non replier | 1196 | 238 |
| replier | 582 | 852 |

**Table 4.11:** Confusion matrix of the classification results using social, topical and activity features for training a logistic regression model. The columns of the confusion matrix show the predicted values and the rows show the reference values. One can see that around 60% of the candidates are classified as non-repliers and only 40% are classified as repliers.

| | all | topical | social | activity |
|---|---|---|---|---|
| $R^2$ | 0.291 | 0.056 | 0.075 | 0.254 |

**Table 4.12:** Goodness of fit of the logistic regression model from Boards.ie measured using the Nagelkerke pseudo R squared.

# Chapter 5

# Discussion of Results

In the following chapter, the results for the two research questions are summarized for Twitter and Boards.ie.

**RQ1: To what extent is communication on social media influenced by social and topical factors?** The results for the Twitter experiments clearly show that social features, which describe the strength of the relation between users, help predicting repliers much stronger than activity features and are more useful than topical features for predicting if a user will reply to another user or not. This suggests that conversations on Twitter are more driven by friendships and social relations rather than topics. The results for Boards.ie show that topical and social features have only a slight effect on the communication behavior. Users in Boards.ie do not have high preferences to reply always to certain users with whom they share similar interests or have a social relation. Intuitively this makes sense in the context of a forum about technical topics, such as the Computer and Technology forum which was used in this work. In these forums users ask questions about technical problems and answer in most cases without preference to certain users.

The best social features for Twitter, were the *NumRepliesRelation*, the *isFriend* and the *FriendsOverlap* features. This suggests that users are far more likely to reply to a message authored by a user who is a friend of them, to whom they have talked in the recent past frequently and with whom they share common friends. Due to the high correlation between the *ReplyPartnerOverlap* and the *FriendsOverlap*, the *ReplyPartnerOverlap* has an almost same effect in predicting repliers than the *FriendsOverlap*. This means that users who share similar communication partners with the author of a message are more likely to reply to it. In Boards.ie there are only two social features calculated, which are the *NumRepliesRelation* and the *ReplyPartnerOverlap*. Due to a high correlation between the *NumRepliesRelation* and the *PostPostagsSimilarity* the *ReplyPartnerOverlap* was neglected, and the *NumRepliesRelation* was the only feature used

in the logistic regression model. The *NumRepliesRelation* has a positive significant effect in the logistic regression model, but improves the accuracy of the logistic regression model only by an F1 score of 0.02 together with the topical features.

Compared to social features in Twitter, topical features which refer to the overlap of topics between author and candidate occurring in the aggregation of a user's Tweets, the aggregation of the names and descriptions of the participated membership list, and in the user profile description, are less useful for predicting repliers. The logistic regression model shows that topical features like the *TweetConceptSimilarity* and the *BioConceptSimilarity* are significantly positively correlated with users' reply probability. This indicates that there is a slight tendency that users who are interested in similar topics are more likely to reply to each other. However, compared to the social features the influence of topical features on the reply prediction is rather small. In Boards.ie the topical feature, represented by the *PostPosttagsSimilarity*, does have only a small influence in the experiment. When using topical features alone in the logistic regression model the average F1-Score accuracy compared to random guesser increases only by 0.05.

The evaluation of the three different topic extraction methods, which are the Alchemy Keyword Extraction, the Alchemy Concept Tagging method and the Twitter POS Tagger, found that the Twitter POS Tagger performs best for both social applications, but brings only a very small improvement in the model fit (increased R squared value of around 0.01). The Twitter POS Tagger extracts the most topics from the most topics the text followed by the Alchemy keyword extraction. The least number of topics are extracted when using Alchemy Concept tagging method.

Finally, the results of the significance analysis show that in Twitter the activity features *AvgTweetActivityLastWeek*, *ReplyActivity* and *Followee* are positively significant. This demonstrates that the activity level of a user is indeed a significant factor, which influences if a user will reply to a message or not. Not surprisingly, active users are more likely to reply than non active users. The *Followee* feature is significantly negative, which means that the more users someone follows the less likely it is that the user reply to a certain message. The features which are related with the popularity and social status of a user (*Openess* and *Followers* ) are not significant in Twitter. In Boards.ie the activity feature *PostActivityLast2Week*, which refer to the number of messages written in the last 2 weeks before the initial message of the conversation was written, has the highest impact in predicting repliers.

**RQ2: To what extent are conversation partners on social media platforms predictable?** This work shows that for Twitter conversations a binary classification model that differentiates between users who will and will not reply to each other may achieve an F1-score of 0.75 using social, topical and activity features. Using topical features as independent variables leads to the worst statistical model, while using a combination of all features works best, followed by using social features alone. It was possible to

increase the average F1 score of a random baseline classifier by 24% when using social features alone. In Boards.ie conversation the classification model achieved an F1 score of 0.71 when using all features. Boards.ie conversations are more dominated by active users, and when using only activity features in the model the F1 score of a baseline classifier is increased by 19%. This illustrates that Twitter and Boards.ie conversations are to a certain amount predictable.

In the case of Twitter an average F1-Score of 0.75 suggests, assuming that precision and recall for both classes are equal, that the likelihood to correctly classify a follower of an author (writing the initial message within a conversation) as a replier or non-replier is around 75%. But one has to keep in mind, that there is a bias in the positive samples which comes from picking a random follower for the candidate in the negative samples. These followers are likely to contain a certain percentage of users who are not active at all in Twitter. In case of restricting the candidate in the negative samples to be an active follower of the author, not only a follower, the F1-Score is expected to be smaller than 0.75. An F1-Score of almost 1.0, which would be a perfect classification, is practically not possible because of the candidates in the positive and negative samples which have almost the same feature values, and therefore a classification model can not discriminate these users.

# Chapter 6

# Conclusions and Implications

This thesis gives a comprehensive insight into the dynamics of Twitter and Boards.ie conversations. In this research, the impact of topical, social and activity features in the prediction of repliers was analyzed as well as, how accurately repliers can be predicted in Twitter and Boards.ie conversation. The results of this thesis illustrate that replying is indeed not a random process and it is shown that repliers can be distinguished from non-repliers up to an F1 score of 75%. The results also show that the importance of social, topical and activity features highly depend on the context of the conversations and the social application in which the conversation takes place. While Twitter conversations are more driven by social features, which refer to the strength of social relation between users, than by topical features (also suggested by Sousa et. al. [6]), Boards.ie conversations are more dominated by active users. The conversations used in this thesis are a snapshot of all conversations occurring in Twitters public timeline and on conversations occurring in the Computer and Technology forum in Boards.ie. Therefore all kind of conversations including social conversations as well as more formal ones were covered in this analysis. When changing the scope, for instance, focusing only on conversations about a special topic, different results can be expected. However, the approach presented in this thesis is general and can be applied to other conversation datasets as well.

One of the main contributions of this work is the definition of a comprehensive set of features to quantify the major social and topical factors which may impact users' communication behavior. Further, this thesis proposes different statistical methods to analyze the impact of each feature on the prediction of repliers.

This work has implications for researchers and practitioners who are interested in understanding conversation dynamics. On the other hand, the prediction of repliers can be incorporated into the design of online conversations. A concrete example can be the incorporation into orchestration systems for video communication where users which have a higher chance to reply to the current speaker can be prioritized on the screen.

## 6.1 Limitations and Future Work

This work has certain limitations since in case of the Twitter experiment it is assumed that all users who follow a user are similar likely to see messages authored by this user, which is a simplification and may not reflect the reality. By adding activity features as covariates this limitation is addressed to some extent. An interesting future investigation could be to evaluate in more detail, how to find users who are likely to see a message. Based on this evaluation, one could reduce the influence of users who are not able to see a certain message and therefore have no chance to reply to it.

This work investigates only the first replier on a single branch within a conversation, and does not take the long-term dynamics of social media conversations into account. Analyzing the long-term dynamics of online conversations can be also an interesting task for further investigations. Further, one also has to point out that any crawling strategy might introduce a certain bias in the data, as comprehensively studied and described in [10].

This work focuses on features which can be computed between pairs of users rather than triples (consisting of the two users and the current message) since the initial motivation was to integrate this work into a real-time video communication tool [15] which exploits users' social media stream as background knowledge for orchestrating the video communication. Therefore, it is necessary to be able to compute the features at the beginning of each communication session rather than re-computing them after each message or sentence. For future work it is interesting to analyze also the influence of the current message on users' reply behavior and update the initial communication prediction model during the course of a conversation.

A problem which occurs in the Boards.ie analysis is that social and topical features are strongly correlating. Therefore it is difficult to make conclusion whether topical or social features have a bigger impact on the prediction of repliers. However, the prediction experiment shows that topical and social features together only increase the average classification accuracy by an F1-Score of 0.02, therefore these features do not have a big impact in the experiment.

The prediction model presented in this thesis is at the moment only able to classify users as repliers, which are users who are likely to reply to a certain message, or non-repliers which are users who are not expected to reply. An interesting extension of this model would be to predict the user with the highest chance to reply to a certain message.

# Appendix A

# Appendix

## A.1 Technological Infrastructure and Tools Used

### A.1.1 R Environment

For the significance analysis, the statistical program R in version 2.15.2 was used. Table A.1 gives an overview of additional R libraries used.

| Library | Description |
|---|---|
| foreign | Functionality for reading and writing data stored by different statistical program. *Usage: Reading data files stored in ARFF format.* |
| corrgram | Is calculating the correlation between variables and plots the correlation matrix. *Usage: Plotting the correlation matrix.* |
| rms | Includes following functionality: Regression modelling, validation, prediction, testing, estimation, graphics and typesetting. *Usage: Calculating the r squared value.* |

**Table A.1:** Overview of Python libraries.

### A.1.2 Python Environment

For the dataset generation and Feature generation, the 64 bit version Python2.7[1] was used. Table A.2 gives an overview of the used python libraries.

[1] http://www.python.org/

| Library | Description |
|---|---|
| Ipython | Is an interactive shell used for python and offers tab completion, introspection, advanced shell commands and rich history |
| NumPy | Allows fast and easy multi dimensional array manipulation. |
| SciPy | Mathematics library for scientific working. It depends on the NumPy library. *Usage: I am using it for calculating the statistical measures such as the mean, median and the standard deviation of vectors.* |
| NLTK | It is a natural language processing library. *Usage: In this work I use it for text pre-processing purposes.* |
| Guess-language | Tool for guessing the language for a specified text. Depends optionally on the spell checking library PyEnchant which improves the classification accuracy for Tweets. *Usage: I use this library for filtering English Tweets in the conversation crawler.* |
| Python-Twitter | TwitterAPI wrapper for python. *Usage: Crawling the Twitter REST API for user information* |
| Tweepy | TwitterAPI wrapper for python. *Usage: Crawling the Twitter Streaming API for conversations.* |
| Alchemyap | API used for extracting high for semantic analysis of text. *Usage: Extracting concepts from text.* |
| Simplejson | A simple JSON encoder/decoded. |
| Mysqldb | MySQL wrapper library to access a MySQL database. |

**Table A.2:** Overview of Python libraries.

## A.2 Twitter Experiment Workflow Implementation

In this section, the implementation of the Twitter experiment is described. The whole workflow was dived in the following steps:

1. Crawling for Twitter conversations.
2. Creating positive and negative samples.
3. Crawling for user information for each user within the positive and negative samples.
4. Feature generation.
5. Creating the correlation plot and the boxplots for all features.
6. Significance Analysis for the features.
7. Prediction Experiment.

### A.2.1 Step1: Crawling for Twitter Conversations

The conversation crawler, written in Python, uses the Twitter Streaming API to crawl the public timeline for reply messages in English language.

The following code snippet shows one of the main parts of the conversation crawler. For each Tweet crawled from the public timeline, it is checked whether the in_reply_to_status_id is set and the language of the Tweet is English (line 1). The code includes also a simple mechanism to prevent the user account to be blocked (line 4 to 7). The crawler is send to sleep when the current API call lmit it less than 300. 300 was used, because I had access to a white-listed Twitter account which allows 20000 API request/hour. If the message is identified as a reply the method CrawlConversation(line 8) is called which recursively crawls for Tweets until the root of the conversation is reached. In line 10 one can see the definition of the CrawlConversation method. The method recursivly calls itself (line 12) as long as the *in_reply_to_status_id* attribute of the current Tweet (status) is not set. This indicates that the start message of a conversation was found.

```
1 if status.in_reply_to_status_id and guess_language(status.text
    )
2 ==”en”:
3   limit =self.api.rate_limit_status()['remaining_hits']
4   while limit < 300:
5     print”sleeping”
6     time.sleep(10)
```

```
7      limit = self.api.rate_limit_status()['remaining_hits']
8     conversation = self.CrawlConversation(status)
9
10 defCrawlConversation(self, status):
11   if status.in_reply_to_status_id:
12     return self.CrawlConversation(self.api.get_status(status.
          in_reply_to_status_id) )+ [status]
13    else:
14     return[status]
```

When crawling Twitters public timeline for conversations, it needed in average around 10 API calls/conversation.

**Conversation file format**

The above mentioned conversation crawler outputs as a result a file with conversations. Each conversation within the file contains n-messages. Each message consists of the following information: TweetID, TweetText, UserID and CreationDate (Y-M-D h:m:s). Conversations are separated using an empty line. The following text is a snapshot of conversations in the created conversation file.

```
1 31231 How are u?  51379912  2012-10-30 10:30:20
2 31232 I am fine 45852399  2012-10-30 10:35:20
3
4 34258 Today is a very nice day :)  51379912 2012-11-30
     14:10:50
5 31232 I can only agree  45852399  2012-11-30 14:12:20
```

### A.2.2 Step2: Creating Positive and Negative Samples

The process to create positive and negative samples is already described in section in detail.

### A.2.3 Step3: Crawling for User Information

For the task of crawling further information about a Twitter user, such as the follower network, the membership lists, the public timeline and the friendship relation between two user, a tool was developed with the following properties:

- For each kind of user information which should be crawled, a entry in a tasks table has to be created. For instance, to crawl the follower list of a specific user, one has to add an entry in the task table first.
- This tasks table is then used by a crawling tool, which gathers the needed data from Twitter using its API.
- To increase the request throughput rate per hour, the crawler is capable to send many request parallel. This is especially important when using a white-listed Twitter API which allows 20000 API requests per hour.
- The crawling tool includes a sleeping mechanism when the request limit is reached.

The crawling tool uses an extended version of the Python TwitterAPI wrapper library *Python-Twitter*[2] to access Twitter's REST API in version 1.0[3]. The crawler uses as input a SQLite database which comprises a $tasks$ table containing an entry for each user information that should be crawled. The structure of the $tasks$ table is illustrated in Table A.3.

The following list shows the API requests needed for crawling user information using the REST API in version 1.0.

- Timeline: 1 request for 20 Tweets within the user's timeline. It is possible to crawl back the last 3200 Tweets within a user timeline.
- Follower: 1 request for max. 5000 followers.
- Followees: 1 request for max. 5000 followers.
- Membership lists: 1 request for 20 lists per user.
- Friendship: 1 request per author- candidate pair.
- User Profile: 1 request per user.

### A.2.4 Step4: Feature Generation

For the calculation of all features values, a *Feature-Generator* tool was created which was used on the one hand to analyze the performance of a single feature, and on other the hand to calculate for each sample in the positve and negative samples the corresponding feature values and, moreover, the tool delivers as a result an *ARFF*[5] file which can be easily importet in statistical programs such as R and Weka.

[2] http://code.google.com/p/python-twitter/
[3] https://dev.twitter.com/docs/api/1
[5] http://www.cs.waikato.ac.nz/ml/weka/arff.html

| Attribute | Description |
|---|---|
| type | Describes the type of user information that should be crawled. It can have the following values:<br>• status: returns the text of a Tweet text using its Tweet id (statuses/show [4] request<br>• profile: returns a user profile using a user id (users/show1 request)<br>• membership: returns the membership list using a user id (lists/memberships1 request)<br>• friendship: returns friendship information between two users using two user ids (friendships/show1request)<br>• friends: returns followees using a user id (friends/ids1 request)<br>• followers: returns followers using a user id (followers/ids1 request)<br>• timeline: returns the public timeline(stream of Tweets written by a user in chronological order) of an user |
| id | Depending on the *type* attribute, the*id* could have the following values:<br>• Tweet_id, needed for the *type* status<br>• User_id: needed for the types *membership, friends, followers and timeline*<br>• [User1_id,User2_id](JSON decoded list of two user ids) needed for the type *friendship* |
| finished | Status of the task [open, closed]. |
| results | Result of the API request. |

**Table A.3:** Structure of the tasks table. This table is used to describe the user information which should be crawled.

To calculate and analyse a single feature the Feature-Generator provides a GUI which allows the definition of a feature using a formula defined in python syntax. For instance to calculate the cosine similarity between two topic vectors, one can use the formula:

$$cosine\_similarity(tweet\_concepts(A), tweet\_concepts(C)) \tag{A.1}$$

As a result the tool would calculate for each sample, represented as triple $< A, C, Class >$, the cosine similarity between the list of concepts extracted from the Tweets written by

user A and the list of concepts from user C. To get an first impression of the discriminative power of this feature, the tool can plot a boxplot (figure A.1) which shows the distribution of all samples within the replier and non-replier class, additionally, it calculates the mean, median and the standard distribution of each distribution.

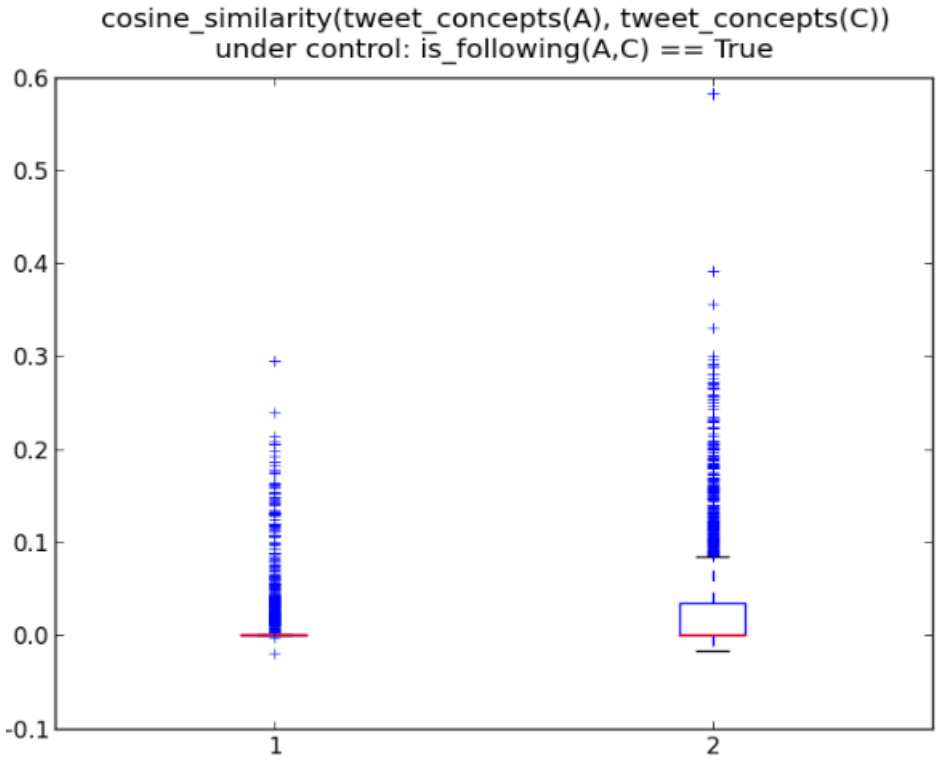


**Figure A.1:** Screenshot of a boxplot plotted by the Feature-Generator. This boxplot shows the distribution of the Tweet concept overlap between A(author) and C(candidate) in both classes.

Figure A.2 shows the GUI of the Feature-Generator tool. The GUI offers the following functionality (the list numbers refer to the numbers in figure A.2):

1. Text field to enter a feature formula in python syntax. The formula will be interpreted as native python code, and is running in a defined context which includes predefined variables and functions (context will be described later in this section).

2. A text field to enter a precondition which must be fulfilled for each sample to calculate the feature formula. The precondition is also interpreted as python code and runs in the same context like the feature formula. In figure A.2 the precondition $is\_following(A, C)$ is fulfilled when user A is following user C.

3. Starts the calculation of the feature formula for each sample.

4. Stops the on-going calculation (useful when the calculation seems to have errors and take longer time)

5. Normalizes the calculated feature values.

6. Plots a boxplot for the calculated feature A.1.

7. Plots the log odds of a feature.

8. Prints basic statistical measures like median, mean, standard deviation and maximum and minimum value of the feature distribution in the replier and non-replier class.

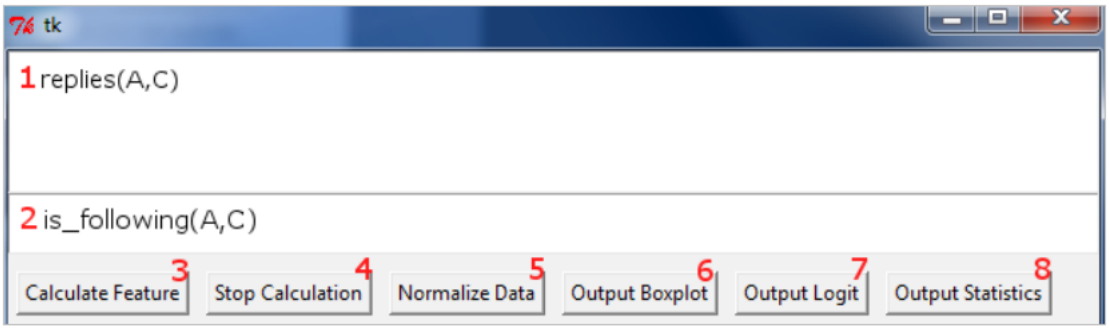


**Figure A.2:** Screenshot of the graphical user interface of the Feature-Generator.

The feature formula and the precondition are executed in a python context which contains a list of predefined variables and functions. Table A.4 gives an overview of all defined variables. While table A.5 gives an overview of all predefined functions to gather user data, tableA.6 lists some useful helper function defined in the python context.

| Variable | Description |
|---|---|
| A | Refers to the $author's$ user_id in one sample |
| C | Refers to the $candidate's$ user_id in one sample |
| CL | Refers to the class (replier or non-replier) of one sample. |

**Table A.4:** Predefined context variables, which can be used in a feature formula when using the Feature-Generator.

In the following, the formulas to calculate all features are listed.

```
#Topical Features
tweet_postags_similarity = cosine_similarity(tweet_postags(A),
    tweet_postags(C))
```

```
3  tweet_keywords_similarity = cosine_similarity(tweet_keywords(A
       ), tweet_keywords(C))
4  tweet_concepts_similarity = cosine_similarity(tweet_concepts(A
       ), tweet_concepts(C))
5  list_postags_similarity = cosine_similarity(list_nd_postags(A)
       , list_nd_postags(C))
6  list_keywords_similarity = cosine_similaritylist_nd__keywords(
       A), list_nd_keywords(C))
7  list_concepts_similarity = cosine_similarity(list_nd_concepts(
       A), list_nd_concepts(C))
8  profile_postags_similarity = cosine_similarity(profile_postags
       (A), profile_postags(C))
9  profile_keywords_similarity = cosine_similarity(
       profile_keywords(A), profile_keywords(C))
10 profile_concepts_similarity = cosine_similarity(
       profile_concepts(A),profile_concepts(C))
11
12 #Social Features
13 common_list_membership = jaccard_index(lists(A),lists(C))
14 cand_in_author_list = jaccard_index(created_lists(A), lists(C)
       )
15 num_replies_relation = replies(A,C)
16 reply_partner_overlap = jaccard_index_dicts(reply_partner(A),
       reply_partner(C))
17 isFriend = is_friend(A,C)
18 friends_overlap = jaccard_index_dicts(
19     average(list_to_dict(friends(A)),list_to_dict(
20     followers(A))),average(list_to_dict(friends(C)),
21     list_to_dict(followers(C))))
22
23 #Activity Features
24 tweet_actvity = num_tweets(C)
25 reply_activity = num_replies(C)
26 avg_tweet_actvity_week = avg_tweets_week(C)
27 num_replier = replyingto(C)
28 followers = num_followers(C)
29 followees = num_friends(C)
```

| Variable | Description | Return Type |
|---|---|---|
| replies(user1, user2) | Number of replies from user2 to user1 | Integer |
| num_tweets(user) | Number of Tweets from a specified user. | Integer |
| num_replies(user) | Number of replies from a specified user. | Integer |
| reply_partner(user) | List of users, a specified user was communicating with. | List |
| followers(user) | List of followers from a specified user. | List |
| followees(user) | List of users a specified user is following. | List |
| tweet_concepts(user) | Extracted Alchemy concepts from the aggregation of Tweets written by a specified user. | Dictionary |
| tweet_keywords(user) | Extracted Alchemy keywords from the aggregation of Tweets written by a specified user. | Dictionary |
| tweet_postags(user) | Extracted POS tags (Twitter POS Tagger) from the aggregation of Tweets written by a specified user. | Dictionary |
| tweet_texts(user) | Aggregated Tweet texts written by a specified user. | String |
| profile_concepts(user) | Extracted Alchemy concepts from the profile description written by a specified user. | Dictionary |
| profile_keywords(user) | Extracted Alchemy keywords from the profile description written by a specified user. | Dictionary |
| profile_postags(user) | Extracted POS tags (Twitter POS tagger) from the profile description written by a specified user. | Dictionary |
| list_nd_concepts(user) | Extracted Alchemy concepts from the names and descriptions of all membership lists a specified user is participating. | Dictionary |
| list_nd_keywords(user) | Extracted Alchemy keywords from the names and descriptions of all membership lists a specified user is participating. | Dictionary |
| list_nd_postags(user) | Extracted POS tags(Twitter POS tagger) from the names and descriptions of all membership lists a specified user is participating. | Dictionary |
| list_nd_text(user) | Aggregated text occurring in the names and descriptions of the membership list a specified user is participating. | String |
| lists(user) | All membership lists a specified user is participating. | List |
| created_lists(user) | All membership lists created by a specified user. | List |

**Table A.5:** Predefined context function, which can be used within a feature formula to gather user data.

| Function | Description |
|---|---|
| jaccard_index(list1, list2) | Calculate the Jaccard coefficient between two lists. |
| similar(list1,list2) | Calculate the number of similar elements occurring in two different lists. |
| cosine_similarity(dict1, dict2) | Calculate the cosine similarity between two python dictionaries. |

**Table A.6:** Predefined helper functions, which can be used within a feature formula when using the Feature-Generator.

### A.2.5 Step5: Correlation Matrix, Boxplots

In this step the correlation matrix and the creation of the boxplots is described. These plots were used in the significance analysis of the Twitter and Boards.ie experiment described in Chapter 4.

The following two code snippets are developed in R[6] and take as input an ARFF file, which includes all positive and negative samples and the corresponding feature values.

The following R snippet calculates the correlation matrix using the Pearson correlation coefficient, and creates a level plot which maps each correlation value to a color.

```
1  library(corrgram)
2  require("foreign")
3  dataset = read.arff("./samples.arff")
4  cor.coef <- cor(as.matrix(dataset[, -c(1,2,3,4,5,6,15,17,
5  length(dataset))]), use="pair", method='pearson')
6  ord <- order(cor.coef[1,])
7  xc <- cor.coef[ord, ord]
8  pdf('./samples_cormat.pdf')
9  par(mfrow=c(1,1), cex.axis=0.6, cex.main=1.0, cex.lab=0.6, xpd
       =TRUE, mar=c(0,0,0,0))
10 print(levelplot(cor.coef,xlab=NULL,ylab=NULL,
11   at=do.breaks(c(-1.01,1.01),101),scales=list(x=list(rot=45)),
12   colorkey=list(space="top"),
13   label.style="align",
14   col.regions=colorRampPalette(c("red","white","blue"))))
15 dev.off()
```

It is important to remove all categorical features (line 3) from the dataset before calculating the Pearson correlation matrix, because it is not possible to calculate the Pearson correlation coefficient between categorical features. For calculating the Pearson correlation coefficient the $cor.coef$ function was used (line 4).

For creating and plotting the feature boxplots, the following code was used:

```
1  require("foreign")
2  dataset =read.arff("./samples.arff")
3  pdf('./samples_feature_boxplots.pdf')
4  par(mfrow=c(3,3), cex.axis=0.8, cex.main=1.0, cex.lab=0.8, xpd
       =TRUE)
5  delete <- c("isFriend", "class")
6  names = colnames(dataset[, !colnames(dataset) delete])
7  for(j in1:(length(names))){
```

[6] http://www.r-project.org/

```
8    boxplot(dataset[, names[j]]~dataset$class,
9      col=rainbow(20),
10     notch=F, main=names[j], names=levels(dataset$class),
11     outline=F, srt=45, horizontal=F,
12     xlab="Class")
13 }
14 dev.off()
```

In line 8 the boxplots are plotted for each feature using the $boxplot$ function in R. The script ouputs a pdf file with all feature boxplots.

### A.2.6 Step6: Significance Analysis

The following R snippets was used for calculating the Wilcoxon rank sum test:

```
1  require("foreign")
2  dataset =read.arff("./samples.arff")
3  res <- data.frame(name=character(0), pvalue=numeric(0))
4  res$name <- as.character(res$name)
5  count <- 1
6  delete <-c("class", "isFriend")
7  names = colnames(dataset[, !colnames(dataset) delete])
8  for(j in1:(length(names))){
9    tmp <- wilcox.test(dataset[, names[j]]~dataset$class)
10   if(tmp$p.value<0.05){
11     res[count, ]$name <- as.character(names[j])
12     res[count, ]$pvalue <- tmp$p.value
13     count <- count+1
14   }
15 }
```

For calculating the Wilcoxon rank sum test for each feature, the R function $wilcox.test$ is used (line 9). The Wilcoxon test can be calculated only for numerical features, therefore the categorical features has to be filtered before (line 6 to 7). To calculate the Chi-squared test for categorical features the $chi2.test$ function in R.

The next R script calculates the logistic regression model using the $lrm$ function. One advantage of using the $lrm$ function is that it automatically calculates the Nagelkerkes pseudo r squared value used evaluating the goodness-of-fit of the logistic regression model.

```
1  require("foreign")
2  dataset = read.arff("samples.arff")
```

```
3 res <- lrm(class ~ Openness + NumRepliesRelation +
     AvgTweetActivityLastWeek + ReplyActivity   +
     ReplyPartnerOverlap  +    TweetKeywordSimilarity +
     Followees + CommonListMembership + BioKeywordSimilarity +
     ListKeywordSimilarity + CandInAuthorsList + isFriend +
     Followers , data = dataset)
4 summary(res)
```

### A.2.7 Step7: Prediction Experiment

To perform the prediction experiment, the data mining program Weka[7] was used. Weka offers many different regression models such as a logistic regression model and makes the use of a 10 fold cross validation easy. Weka uses the ARFF data format for input files.Weka outputs several accuracy measures such as the F1-score, the precision and recall and also plots a confusion matrix which shows the classification results. Figure A.3 shows a screenshot of the Weka GUI.

[7] http://www.cs.waikato.ac.nz/ ml/weka/index.html

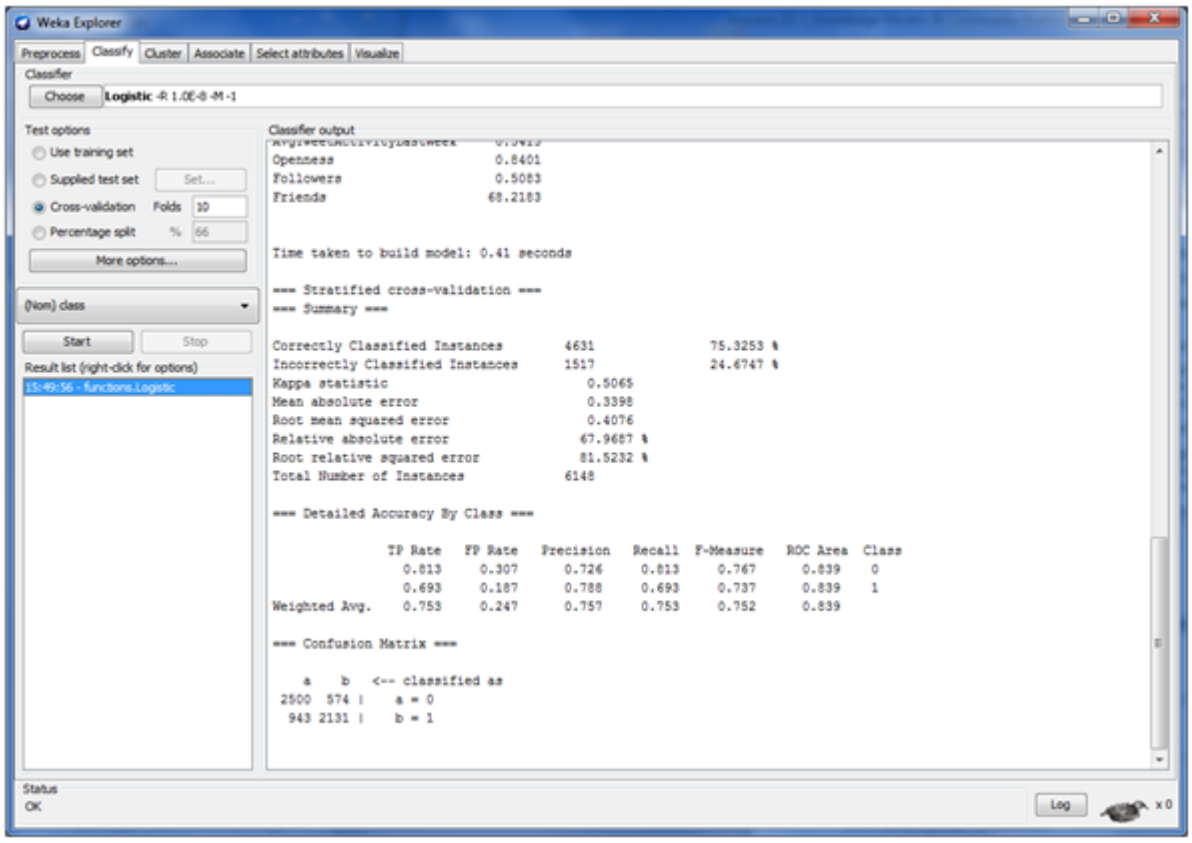


**Figure A.3:** Screenshot of the datamining program Weka.

# Bibliography


[1] Alexa. Top sites. http://www.alexa.com/, 2012. (Cited on pages 1 and 7.)

[2] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, Beijing, 2009. (Cited on pages vii, 20 and 21.)

[3] Charalampos Chelmis and Viktor K. Prasanna. Predicting communication intention in social networks. In *Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust*, SOCIALCOM-PASSAT '12, pages 184–194, Washington, DC, USA, 2012. IEEE Computer Society. (Cited on page 14.)

[4] Jilin Chen, Rowan Nairn, and Ed Chi. Speak little and well: recommending conversations in online social streams. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 217–226, New York, NY, USA, 2011. ACM. (Cited on pages 13 and 36.)

[5] Peter Cogan, Matthew Andrews, Milan Bradonjic, W. Sean Kennedy, Alessandra Sala, and Gabriel Tucci. Reconstruction and analysis of twitter conversation graphs. In *Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research*, HotSocial '12, pages 25–31, New York, NY, USA, 2012. ACM. (Cited on page 32.)

[6] Eduarda Mendes Rodrigues Daniel Sousa, LuÃs Sarmento. Characterization of the twitter @replies network: Are user ties social or topical? 2010. (Cited on page 69.)

[7] Eric Gilbert. Predicting tie strength in a new medium. In Steven E. Poltrock, Carla Simone, Jonathan Grudin, Gloria Mark, and John Riedl, editors, *CSCW*, pages 1047–1056. ACM, 2012. (Cited on page 13.)

[8] Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 211–220, New York, NY, USA, 2009. ACM. (Cited on pages 12 and 14.)

[9] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-speech tagging for twitter: annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 42–47, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. (Cited on pages vii, 22 and 27.)

[10] Sandra Gonzalez-Bailon, Ning Wang, Alejandro Rivero, Javier Borge-Holthoefer, and Yamir Moreno. Assessing the Bias in Communication Networks Sampled from Twitter. *Social Science Research Network Working Paper Series*, December 2012. (Cited on page 70.)

[11] Courtenay Honeycutt and Susan C. Herring. Beyond microblogging: Conversation and collaboration via twitter. In *Proceedings of the Forty-Second Hawai'i International Conference on System Sciences (HICSS-42). Los Alamitos, CA.*, pages 1–10, Los Alamitos, CA, USA, 2009. IEEE Computer Society. (Cited on pages vii, 9, 10 and 31.)

[12] David W. Hosmer and Stanley Lemeshow. *Applied logistic regression (Wiley Series in probability and statistics)*. Wiley-Interscience Publication, 2 edition, 2000. (Cited on pages iii, 15, 16, 17, 18 and 19.)

[13] Bernardo A. Huberman, Daniel M. Romero, and Fang Wu. Social networks that matter: Twitter under the microscope. *CoRR*, abs/0812.1045, 2008. (Cited on page 12.)

[14] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, WebKDD/SNA-KDD '07, pages 56–65, New York, NY, USA, 2007. ACM. (Cited on pages 1 and 8.)

[15] Rene Kaiser, Wolfgang Weiss, Manolis Falelakis, Spiros Michalakopoulos, and Marian Florin Ursu. A rule-based virtual director enhancing group communication. In *ICME Workshops*, pages 187–192. IEEE, 2012. (Cited on pages 1 and 70.)

[16] David Lane. Online statistics education:a multimedia course of study. `http://onlinestatbook.com/`, 2013. (Cited on page 43.)

[17] Sofus A. Macskassy. On the Study of Social Interactions in Twitter. 2012. (Cited on pages 9 and 33.)

[18] Peter Monge and Noshir Contractor. *Theories of Communication Networks*. Oxford university Press, 2003. (Cited on page 2.)

[19] Mor Naaman, Jeffrey Boase, and Chih-Hui Lai. Is it really about me?: message content in social awareness streams. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, CSCW '10, pages 189–192, New York, NY, USA, 2010. ACM. (Cited on pages iii, vii, 10 and 11.)

[20] Nielson. Nielsen's u.s. social media survey 2012. http://www.nielsen.com/us/en/reports/2012/state-of-the-media-the-social-media-report-2012.html, 2012. (Cited on page 1.)

[21] Sharoda A. Paul, Lichan Hong, and Ed H. Chi. Is twitter a good place for asking questions? a characterization study. In Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, editors, *ICWSM*. The AAAI Press, 2011. (Cited on page 33.)

[22] Matthew Rowe, Milan Stankovic, and Harith Alani. Who will follow whom? exploiting semantics for link prediction in attention-information networks. In *International Semantic Web Conference (ISWC)*, volume 7649 of *Lecture Notes in Computer Science*, pages 476–491. Springer, 2012. (Cited on page 11.)

[23] Johannes Schantl, Rene Kaiser, Claudia Wagner, and Markus Strohmaier. The utility of social and topical factors in anticipating repliers in twitter conversations. In *Proceedings of the 5th Annual ACM Web Science Conference*, WebSci '13, pages 376–385, New York, NY, USA, 2013. ACM. (Cited on page 3.)

[24] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002. (Cited on page 20.)

[25] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. (Cited on page 22.)

[26] Twitter. Rest api v1. https://dev.twitter.com/docs/api/1, 2013. (Cited on page 7.)

[27] Twitter. The streaming apis. https://dev.twitter.com/docs/streaming-apis, 2013. (Cited on page 7.)

[28] Twitter. Using hashtags on twitter. https://support.twitter.com/articles/49309, 2013. (Cited on page 6.)

[29] Twitter. Using the twitter search api. https://dev.twitter.com/docs/using-search, 2013. (Cited on page 7.)

[30] Claudia Wagner, Vera Liao, Peter Pirolli, Les Nelson, and Markus Strohmaier. It's not in their tweets: Modeling topical expertise of twitter users. In *Proceedings ASE/IEEE International Conference on Social Computing (SocialCom2012)*, 2012. (Cited on page 48.)

[31] Lei Yang, Tao Sun, Ming Zhang, and Qiaozhu Mei. We know what @you #tag: does the dual role affect hashtag adoption? In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 261–270, New York, NY, USA, 2012. ACM. (Cited on page 12.)

[32] Yao Zhang, Yang A.; Wu and Quing Yang. Yang. 2012. (Cited on page 12.)