TweetSense: Recommending Hashtags for Orphaned Tweets by Exploiting Social Signals in Twitter

by

Manikandan Vijayakumar

A Thesis Presented in Partial Fulfillment
of the Requirement for the Degree
Master of Science

Approved July 2014 by the
Graduate Supervisory Committee:

Subbarao Kambhampati, Chair
Huan Liu
Hasan Davulcu

ARIZONA STATE UNIVERSITY

August 2014

ABSTRACT

Twitter is a micro-blogging platform where the users can be social, informational or both. In certain cases, users generate tweets that have no "hashtags" or "@mentions"; we call it an orphaned tweet. The user will be more interested to find more "context" of an orphaned tweet presumably to engage with his/her friend on that topic. Finding context for an Orphaned tweet manually is challenging because of larger social graph of a user , the enormous volume of tweets generated per second, topic diversity, and limited information from tweet length of 140 characters. To help the user to get the context of an orphaned tweet, this thesis aims at building a hashtag recommendation system called TweetSense, to suggest hashtags as a context or metadata for the orphaned tweets. This in turn would increase user's social engagement and impact Twitter to maintain its monthly active online users in its social network. In contrast to other existing systems, this hashtag recommendation system recommends personalized hashtags by exploiting the social signals of users in Twitter. The novelty with this system is that it emphasizes on selecting the suitable candidate set of hashtags from the related tweets of user's social graph (timeline).The system then rank them based on the combination of features scores computed from their tweet and user related features. It is evaluated based on its ability to predict suitable hashtags for a random sample of tweets whose existing hashtags are deliberately removed for evaluation. I present a detailed internal empirical evaluation of TweetSense, as well as an external evaluation in comparison with current state of the art method.

To my Mom, Dad and Almighty.

ACKNOWLEDGEMENTS

First and Foremost, I would like to express my gratitude to Dr. Subbarao Kambhampati, for his guidance and support as an advisor. His excellent advising method combining guidance and intellectual freedom made my Masters experience both productive and exciting. His constructive criticism and in-depth technical discussions made me rethink my ideas and come up with better ideas. I am extremely thankful for his immense patience in evaluating my research and motivating me to strive for excellence. I believe the skills, expertise and wisdom he imparted on me will significantly enable all my future career endeavors.

I would like to thank my committee members Dr. Huan Liu and Dr.Hasan Davulcu for this guidance and support for my research and dissertation. Their exceptional cooperation and guidance significantly enabled a smoother dissertation process and contributed towards the higher quality of my research. My big thanks to my collaborators Sushovan De and Kartik Talamadupula for the countless hours of technical discussions, contributions made and mentoring me by spending their sweet time between their dissertation and conference deadlines. I would like to thank Yuheng Hu for his suggestions and comments during all phases of my work. I would also like to thank my Fellow Yochanites - Srijith Ravikumar, Tuan Anh Nguyen, Lydia Manikonda, Tathagata Chakraborti, Raju Balakrishnan, Anirudh Acharya, Vignesh Narayanan and Yu Zhang for taking time out of their research to spend time on discussing my ideas and providing me guidance when I needed it the most. I am thankful to my friends Ashwin Rajadesingan and Arpit Sharma for their technical suggestions and comments.

I am grateful to Dr. Erik Johnston and Dr. Ajay Vinze for their support as a Graduate Research Assistant and funding me all the way through my studies. I would like to thank many other faculty members in Arizona State University, without their courses and instruction this dissertation would not have been possible.

My special thanks to my mother (Pushpavalli), father(Vijayakumar) for being my strength and encouraging me with their love and affection throughout my life and education. I specially thank my best friend Arunprasath Shankar, who supported me in all my situations and continue doing. Further, I would like to thank all my friends, classmates, fellow researchers who made my graduate school a better experience.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Twitter is a micro-blogging platform where the users can be social, informational or both. Twitter is more than the sum of its 200 million tweets; it's also a massive consumer of the web itself. People use Twitter for breaking news and content discovery, according to Deutsche's charts [1] . In other words, it has grown beyond the status updates that twitter initially envisaged. As for the motivations of users to actively participate in the Twitter network, Java et. al. [42] identified the following intentions of users such as daily chatter, conversations, information sharing and news reporting.

Figure 1.1: Example Tweet Where the User Comments About Orphaned Tweets

As it is important to know why people use Twitter, it is also worth to understand why people quit Twitter. Many twitter users have commented on how noisy Twitter is Figure 1.1 : That once you follow more than about fifty or so users, your feed becomes unmanageable [18]. If you follow hundreds, it's simply impossible to extract value from

your stream in any structured or consistent fashion. On average, the user's feed gets a few hundred new tweets every ten minutes. It is hard to make sense out of that unassisted. Also users do not seem to be able to find the breaking news and interesting content they want, or even if they think they can, information overload prevents them from getting to it.

As stated by the reporter John McDuling from a article in quartz, Twitter is aware of these issues. The company has described the hashtags and @mentions [11] are used to provide additional context for arcane tweets. Initially, at its launch in March 2006, Twitter did not have hashtags [23] and its users invented these peculiar conventions. Hashtags, words or phrases prefixed with a pound sign #, is the primary way in which Twitter users organize the information they tweet and use it as a metadata tag for the tweet. The #hashtag usage has been evolving since 2007 [3]. Both @mentions and hashtags provide enough contexts for a tweet. Users tend to use different hashtags to refer to same context in their tweets. But the problem is not completely resolved as not all users in twitter use hashtags with their tweets. Based on the data crawled by Eva et. al [60], 49,696,615 tweets out of a total of 386,917,626 crawled tweets contain hashtags, which are approximately 12.84% of all tweets. Also the data that I crawled for my experiment in year 2014 that includes 7,945,253 tweets had 76.30% tweets without hashtags and 23.70% with hashtags. This relatively low percentage already signifies that hashtags are not well-adapted by Twitter users. In regards to hashtags, 87.16% of all tweets do not feature any information about the category or content.

1.1 Motivation

Twitter had 241 million monthly active users at the end of 2013, compared with 1.2 billion for Facebook and 200 million for Instagram [11]. Twitter is also growing more slowly than its peers. Cowen & Co predicts that Twitter will reach only 270 million

monthly active users by the end of 2014, and that the network will be overtaken by Instagram, which Cowen expects to have 288 million monthly active users by then [11]. This significantly low active user from 2013 to 2014 clearly says there are unhappy users in Twitter. The possible reason could be due to noise in Twitter. There are several reasons for noise in Twitter, like the user may omit the hashtags which posting a tweet, user may use incorrect hashtags and user may use many hashtags to get more audience in which some of the hashtags used as a side remarks might not have real context. Some of the articles like the Washington post [19] also emphasize this by saying "90% of a typical twitter feed is basically a waste of everyone's time". There are three different problems exist as stated in the reason for Twitter noise such as Missing hashtag problem, Incorrect hashtag problem and Multiple hashtag usage. In the missing hashtag problem, the user may not use hashtag and this would lead to having fewer contexts in the tweets and the tweet browsers will be subjected to context-less updates. In the case of incorrect hashtag problem, user adds irrelevant hashtags to the topic of the tweet to get global reach. This leads to the scenario of showing irrelevant tweets to the users who are doing content discovery using Twitter search. In the case of multiple hashtag usage, usually the newbies use hashtag spamming to get global reach. Even Twitter emphasized not to over-tag the hashtags as best practices. In this thesis, I will be focusing on the missing hashtag problem as hashtag are supposed to provide additional context in this scenario. The importance of having a hashtag in a tweet is to provide context or metadata for arcane tweets, organize information in the tweets, find latest trends and get more audience. Also, several other articles and user comments mentioned in Figure 1.1 states there is a need for a system to reduce noise by providing additional context. This thesis details the efforts on building a hashtag rectification system for Twitter using its social graph to enhance active user engagement. By being present and active on Twitter, it is possible to create a customized, interactive, information stream with updates from experts in different field, peers and

colleagues across various disciplines, news agencies, and numerous other sources. There is an expectation on Twitter that those who post are interested and available to interact with you; therefore, it is an excellent forum for engaging in continued dialogue about important and timely issues. Twitter's utility grows as one's network grows; and one's network grows the more they interact with others. This in turn would also impact twitter to maintain its monthly active online users in its social network. Further, TweetSense can also be used to reduce noise in Twitter in terms of rectifying hashtags. The impact on solving this problem would be it helps the users to identify the context, help resolve named entity problem, aggregate tweets from users who doesn't use hashtag for opinion mining and do sentiment evaluation on topics.

Figure 1.2: Example Orphaned Tweet

1.2 Problem Statement

The research problem is derived from a scenario when a person who he/she is following in Twitter generate context-less tweets; we call it an Orphan tweet, which have no "hashtags" or "@mentions". Figure 1.2 shows an example of an Orphan tweet. Some of the other examples of Orphan and Non-Orphan tweets from Twitter are listed in Figure 1.3 . If a person wants to "engage" with his/her friend on a topic, presumably he/she will be more interested to find out the context of the Orphan tweet. But getting the context for an Orphan tweet manually is challenging given that the average length of a single

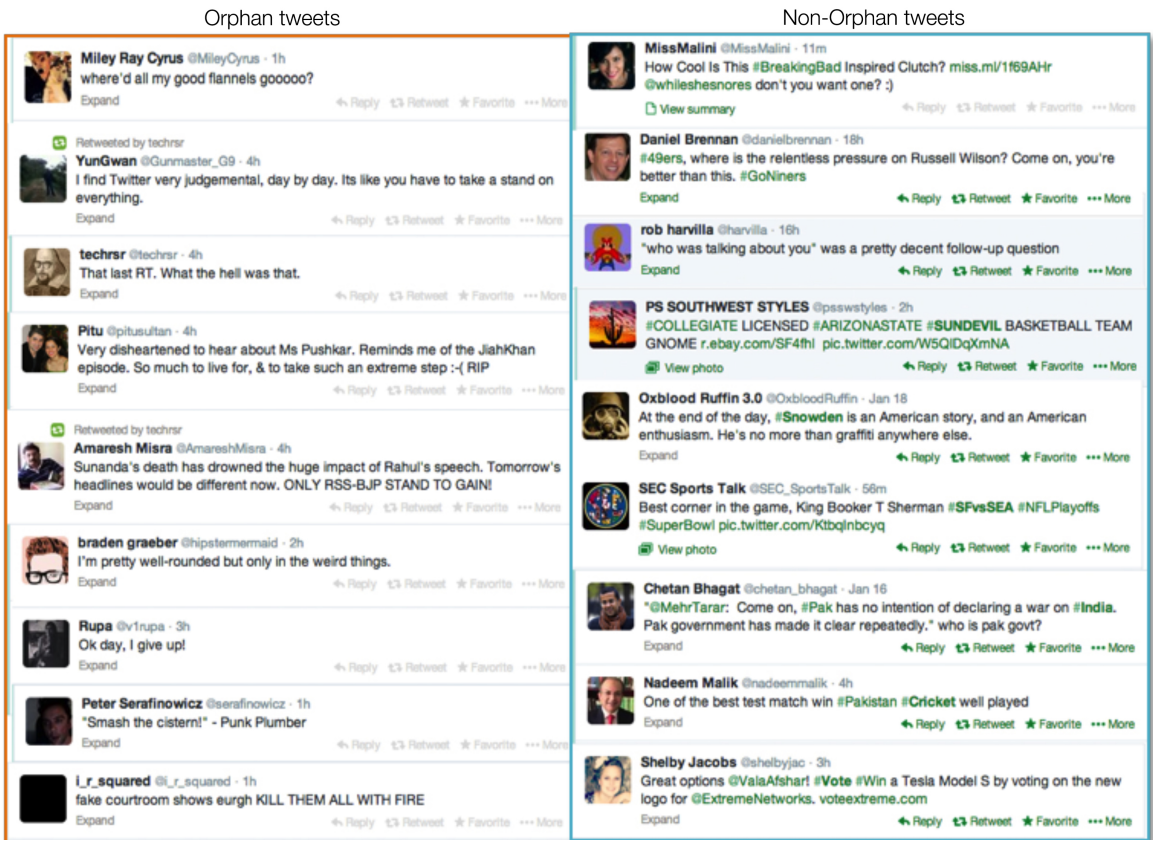


Figure 1.3: Orphan and Non-Orphan Tweets

post is about 14 words or 78 characters [33], which may not provide sufficient information compared with other types of social media [31]. It also has the constraints with enormous amount of tweets tweeted per day and user's larger social graph. As a hashtag acts as a metadata tag for a tweet, we will assume that the context of a tweet means a hashtag. So, building a system that can make better recommendation of hashtags to a tweet would help find the context of a tweet. Existing hashtag recommendation systems [60], [59], [30], [34] are mostly focused on improving twitter search capabilities.

1.3 Proposed Approach

In response to the shortcomings of the current methods as discussed in the previous section, in solving the noise in Twitter streams to help increase the social engagement

and interaction between users in Twitter, in this thesis I propose the system called TweetSense (derived from the context of making sense out of tweets), a hashtag rectification system for tweet browsers in Twitter. My high level idea of rectifying hashtags might look similar to the existing hashtag recommendation system proposed by Eva et. al. [60] which recommend hashtags based on text similarity, recency and global trendiness of the related tweets from the global twitter space. However, TweetSense works for tweet browsers rather for tweet originators. Tweet originator is users who post tweets and tweet browsers are the users who look for additional context and information in tweets. TweetSense does not force users to use the hashtags at the time of origin rather recommend a list of hashtags for the users who look for context. Also, the realization of the baseline system's idea is less effective due to two main challenges: (i) consider the global twitter space on choosing the candidate tweets for computation rather optimally looking at the social graph of a particular user. (ii) Ignoring the social signals like @mentions, favorites, etc., and tie strength of the users in the social network while recommending the hashtag. Further the desirable speed-accuracy tradeoffs are different in rectification system vs recommendation system. Compared to the system proposed by Eva et.al., my system need more time than the recommender system as TweetSense need to compute the compute the feature vectors for all the tweet hashtags in the candidate set, and your system finds the probability that each of the hashtags are the correct hashtag for the query tweet). But the computation can be made faster for each user as the social features remains constant for most of the candidate set of tweets and it varies temporally and based on new friends and followers.The main contribution of TweetSense is to effectively address these two major challenges listed before and rectifying the noise present in the tweets.

Instead, looking into the global twitter data, I came up with a generative model for my system that captures the most relevant data from a user's social graph to rectify hashtags. I define the generative model for my system as when a user uses a hashtag to define the

context for his/her tweet, it is most likely that the user might reuse the hashtags that he/she sees from his timeline. This includes the user's previous tweets along with the tweets created by his friends he/she follows. The user is most likely to use the hashtags, which are temporally close enough. User is also more likely to reuse the tweets from the users whose tweet are favorite, retweeted and @mentioned.

In TweetSense, I try to learn a statistical model that captures the social signals of a user along with a set of tweet content related features to predict the hashtags. The set of social signals are determined to compute the tie strength [55] between the users. The social signals that I used for modeling are the number of mutual friends shared by a user, number of mutual followers shared by a user and closeness of a user based on reciprocal following. I also use the temporally based social signals like number of mutual hashtags used by a user at a particular time frame, number of retweets and tweet favorites that were made back and forth by the user and number of conversations the user has with his friends in a specific time window. The set of tweet related features that I use in my system are the similarity between the tweet text, recency of the tweet to find which are temporally close enough and the trendiness of the hashtag with in the user's social graph.I implemented and evaluated the TweetSense by accessing the Twitter Streaming API "Sprintzer" (which allows crawling upto 1% of all public tweets).

TweetSense is a useful tool for all users in twitter social network to find additional context in terms of hashtags for the orphaned tweets to get engage in conversation with his/her friends and exploit the complete usability of the twitter ecosystem. It also helps to reduce the noise in Twitter by rectifying the incorrect, irrelevant hashtags created by users. I present a detailed internal empirical evaluation of TweetSense in several experiments designed by me; as well do an external evaluation in comparison to the current state of art method [9]. My evaluation is done on a random set of tweets whose existing hashtags are deliberately removed and considered as a ground truth for evaluation.

1.4 Organization of Thesis

In Chapter 2, I give an overview of related work. In Chapter 3, I describe the basic algorithm/workflow of my system, list of features that I considered for my system, reason to choose the features. In Chapter 4, I show my ranking methods on how the features are computed for a input qyery tweet without hashtag. In Chapter 5, I describe how I combine the feature scores using a logistic regression model.I then discuss the experimental setup in Chapter 6. Chapter 7 presents my evaluation, and various results that validate my hypotheses. I conclude with an overview in Chapter 8.

Chapter 2

RELATED WORK

Although recommendation of hashtags for online resource has been a popular research area since the spread of web 2.0 paradigms, Traditional recommendation systems are more focused on enhancing the search capabilities and quality.Apart from information retrieval challenges, some of the existing model [24] is more focused on enhancing opinion mining and text classification. The difference between tag recommender systems and traditional recommender systems is that traditional recommender systems are based on two dimensions: users and items based on which recommendations are computed. Tag recommender systems add another dimension, namely tag. The recommendation of tags of online resources like images, bookmarks or bibliographic entries is directly related to my approach. Such approaches can be based on the co-occurrence of tags, like e.g. in [48, 50]. The notion of co- occurrence of tags describes the fact that two tags are used to tag the same photo. Therefore, only partly tagged photos can be subject to tag recommendations. Based on these relatively simple approaches, the paper by Rae et al. [53] proposes a method for Flickr tag recommendations, which takes different contexts into account. However, the task of recommending traditional tags differs considerably from recommending hashtags. My recommendation is solely based on the tweet and its related features in Twitter whereas in traditional tag recommender systems, much more data is taken into consideration for the computation of tag recommendations. As Twitter is a dynamic platform where new hashtags keep evolving around trending topics, the recommendations have to consider this dynamic nature of Twitter. The recommendation of Twitter hashtags can benefit from various other fields of research. These areas are tagging of online resources, traditional recommender systems, social network analysis and

Twitter analysis. As for the recommendation of items within Twitter or based on Twitter data, there have been numerous approaches dealing with these matters. Hannon et al. [36] propose a recommender system, which provides users with recommendations for users who might be interesting to follow. Chen et al. present an approach aiming at recommending interesting URLs to users [27]. The work by Phelan, McCarthy and Smyth [52] is concerned with the recommendation of news to users.

Kwak et al. [45] did a thorough analysis of the Twitter universe focusing on information diffusion within the network. There have been numerous papers throughout the last years addressing the social aspects of Twitter and social online networks in general. Huberman et al. [39] found that the Twitter network basically consists of two networks: one dense network consisting of all followers and followees and one sparse network consisting of the actual friends of users. Huberman defines a friend of a user as another Twitter user with whom the user exchanged at least two directed messages. Boyd et. al. [22] contains an analysis of the retweet messages and Honey et. al. [38] is concerned with how Twitter might be suitable for collaboration by exchanging direct messages. Evandro Cunha et. al. [28] present a first description of gender distinctions in the usage of Twitter hashtags. Men and women use language in different ways, according to the expected behavior patterns associated with their status in the communities.

Some of the recommendation system that share similarity with my approach in a higher level idea of recommending hashtags for tweets includes [21], [26], [32], [46] and [56]. Mesharay et. al [21] system discusses taking history tweets in his/her mother language and predicting interests after he has moved to his/her new location. TeRec [26] extend the online ranking technique and propose a temporal recommender system. Wei Feng [32] developed a statistical model for Personalized Hashtag Recommendation considering rich auxiliary information like URLs, locations, social relation, temporal characteristics of hashtag adoption, etc. Monky we et. al [46] propose a novel hashtag

recommendation method based on collaborative filtering and the method recommends hashtags found in the previous month's data. They consider both user preferences and tweet content in selecting hashtags to be recommended.

The current state of art system that I am referring as a baseline for my system is proposed by Eva et. al. [60] who present an approach for the recommendation of hashtags suitable for the message the user currently enters which aims at creating a more homogeneous set of hashtags. Most of the recommendation system listed above focuses more on the relevance, creditability and co-occurrences. To the best of my knowledge, recommending hashtags as a context for a context-less tweet by using the user's history and social signals has not been attempted. Recommending hashtags based on tweet similarity and recency was attempted by Eva et. al [60] but they haven't considered the approach on utilizing the user's background information into account. I expect my hashtags are recommended based on the tweet related and user related features such as similarity, recency, mutual friends, followers, common hashtags and conversations. Further I base my recommendation process on the hypothesis that users tend to reuse hashtags they already made use of, an analysis of the hashtags previously used by the user and a subsequent incorporation of these hashtags in my recommendation process. I use the recommended hashtags to rectify the tweets with missing hashtags. Though in a higher level my proposed approach might look like a hashtag recommendation system which focuses on Twitter originators, my system focus more on twitter browsers who look for additional context of orphaned tweets. To be more specific, my system is a rectification system for missing hashtags problem rather a recommendation system as the existing state-of-art system.

Chapter 3

HASHTAG RECTIFICATION PROBLEM

The research problem that I am focusing on is, when a user gives a tweet without a hashtag along with the information about the user who posted the tweet as an input to my system, it should be able to recommend a most likely hashtag as a context for that input query tweet. At a higher level, I modeled my research problem as a hashtag rectification system which recommends hashtags as a context for the input tweets. To set up the model, the first step is to track down K most promising hashtags from the dataset, which are basically the ones that have the probability of P(h|u) over a threshold where h=hashtag and u=user. The candidate dataset is derived from the generative model of my system as shown in Figure 3.1 by tacing down the user's timeline.When I mention Twitter user's timeline in the description of my generative model, it is a long stream showing all tweets from those you have chosen to follow on Twitter as defined in Twitter documentation [6].The newest updates are at the top .

In the current state-of-art [60], tweets without hashtags are not considered for the computation of hashtag recommendation. But in my system I make use of all tweets from the user's timeline. I use the tweets without hashtags to compute user's social signals to determine the most influential user for the user who posted the orphan tweet.And I use the tweets with hashtags to determine the hashtags into which the user made tweets around a particular time window similairty with their tweet text and globally trending hashtags.

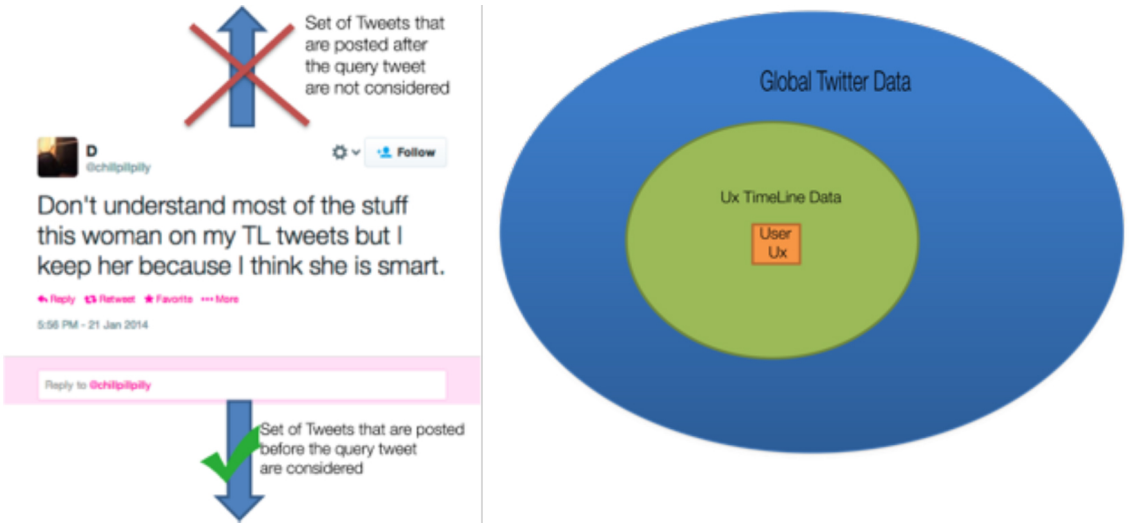


Figure 3.1: Choosing Dataset and Tracing Down K Most Promising Hashtag

3.1 Basic Algorithm

In Figure 3.2 the algorithm and workflow underlying the computation of hashtag recommendations is depicted. My algorithm accepts two parameters as inputs, the query tweet and the user U_x who tweeted the query tweet. After the given initialization steps, tweets from the user U_x timeline is collected. This is realized by indexing the tweets in prior to executing the algorithm. In order to compute the ranking of the hashtag, I extract the important attributes present in the tweets. The attributes such as hashtags, tweet text, temporal information, social signals such as @mentions, favorites, retweets, mutual friends, mutual followers and mutual hashtags. To combine all the scores, I learn the weights for each variable through a statistical model based on their tweet and ground truth hashtag pair. I then use the model to predict the top K hashtags and rank the hashtags based on the confidence of their prediction probabilities. The final lists of ranked hashtags are presented to the user as the context for the orphan tweets.

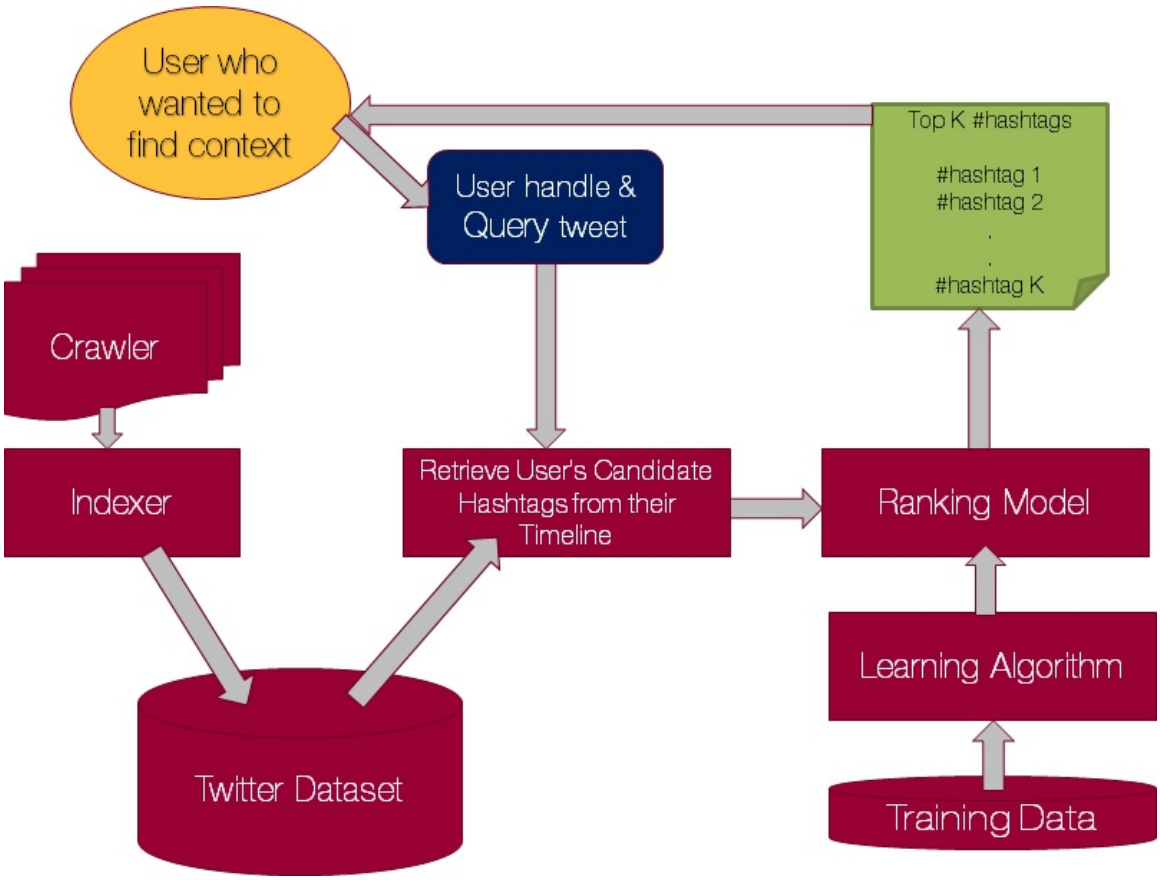


Figure 3.2: TweetSense Architecture (Modified Source:Wikipedia)

3.2 List of Features

I categorize the list of features that will be used by TweetSense into Tweet content related and User related features. All the list of features are briefly explained in the following sections.

3.2.1 Tweet Content Related Features

Tweet Text

Tweets are the basic atomic building block of all things Twitter. Users tweet are also known more generically as "status updates." Tweets can be embedded, replied to, favor-

ited, unfavorited, deleted etc., as stated by Twitter documentation [16]. The maximum length of such a message is 140 characters. Figure 3.3 shows an example tweet tweeted by Jack Dorsey.

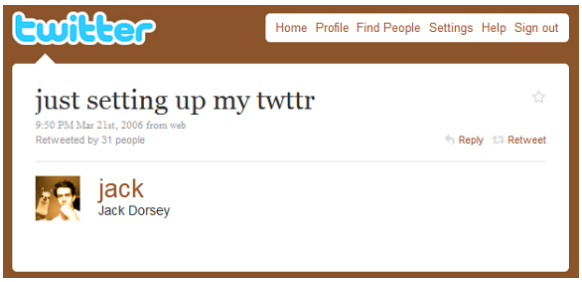


Figure 3.3: Example Tweet

Temporal Information

When a tweet is posted by a user it holds the information about the time the tweet was created. The created at time is in UTC(Coordinated Universal Time) time format. It holds the information about the time in terms of hours, minutes and seconds along with the details of date, month, year and day. An example created at time format obtained from the Twitter API looks as follows "created_at":"Wed Aug 27 13:08:45 +0000 2008"

Trends

Trending topics are those topics being discussed more than others. As Twitter explains trending topics, "Twitter Trends are automatically generated by an algorithm that attempts to identify topics that are being talked about more right now than they were previously. The Trends list is designed to help people discover the most breaking news from across the world, in real-time. The Trends list captures the hottest emerging topics, not just what's most popular. [15]

3.2.2 User Related Features

Retweet

Figure 3.4: Example ReTweet

A Retweet is a re-posting of someone else's Tweet. Twitter's Retweet feature helps you and others quickly share that Tweet with all of your followers. Sometimes people type RT at the beginning of a Tweet to indicate that they are re-posting someone else's content. This isn't an official Twitter command or feature, but signifies that they are quoting another user's Tweet. According to the recent modification in Twitter, Retweets look like normal Tweets with the author's name and username next to it, but are distinguished by the Retweet icon and the name of the user who retweeted the Tweet. You can see Retweets your followers have retweeted in your home timeline. Retweets, like regular Tweets, will not show up from people you've blocked [15]. Boyd et. al. [22] inspected the retweeting behavior of users in Twitter. They state that users make use of retweets as a form of both information diffusion and also the participation in a diffuse conversation. As such, users retweet in order to engage with others about certain topics. The authors found that retweeting users fall into two categories: preservers and adapters. Preservers are users who retweet messages without editing the original message whereas adapters edit a message before retweeting it. Figure 3.4 shows an example of a retweet

Favorite

A 'Favorite' on Twitter refers to topics or subjects that users are most interested in. Every user in social media websites is unique and this is why it's important for the social media

sites to identify their particular interests. Favorites, represented by a small star icon next to a Tweet, are most commonly used when users like a Tweet. Favoriting a Tweet can let the original poster know that you liked their Tweet, or you can save the Tweet for later. [7]. Favorites are described as indicators that a tweet is well liked or popular among online users. When you mark tweets as Favorites, you can easily find useful and relevant information that you can refer back to in the future. You can further share these to friends and other online contacts. [20]

@ Replies and Mentions

Every time your username is tagged on Twitter with the @ symbol (and assuming you haven't blocked the user), it works its way to your mentions folder (which is located under the Connect tab on Twitter.com). A reply is a response in the form of a post to another user, usually to answer a question or in reaction to an idea that has been posted. To reply, the user type in the '@' sign followed by the username, i.e. @username and then follow with your message. Because @replies are different than simple mentions, people (and organizations) can have conversations with one another without cluttering the home feed of those who are only interested in one of the two accounts. [5]. Figure 3.5 is an example of replies.

Figure 3.5: Example of Replies

Mentions used to be known as 'replies' in earlier days of Twitter, this referred to tweets that started with a @username [12]. A mention is not necessarily a direct response to another user and is mostly applied as an FYI(For Your Information). It is placed anywhere in the body of the tweet, not at the beginning, i.e. It's a great day today @username. [13].If a given @username is included in a tweet anywhere else but at the very start, Twitter interprets this differently as a mention instead of a reply. Put literally anything ahead of the @ symbol on a tweet and it isn't a reply. Figure 3.6 is an example of @mentions.

Figure 3.6: Example of Atmentions

Hashtags

Twitter is the birthplace of modern hashtag usage as such, its hashtags are more versatile than other sites. The # symbol, called a hashtag, is used to mark keywords or topics in a Tweet. It was created organically by Twitter users as a way to categorize messages [9]. People use the hashtag symbol # before a relevant keyword or phrase (no spaces) in their Tweet to categorize those Tweets and help them show more easily in Twitter Search. Clicking on a hashtagged word in any message shows you all other Tweets marked with that keyword. Hashtags can occur anywhere in the Tweet at the beginning, middle, or end. Hashtagged words that become very popular are often Trending Topics. Figure 3.7 shows an example tweet tweeted by Chris Messina who proposed the usage of Hashtag in Twitter.

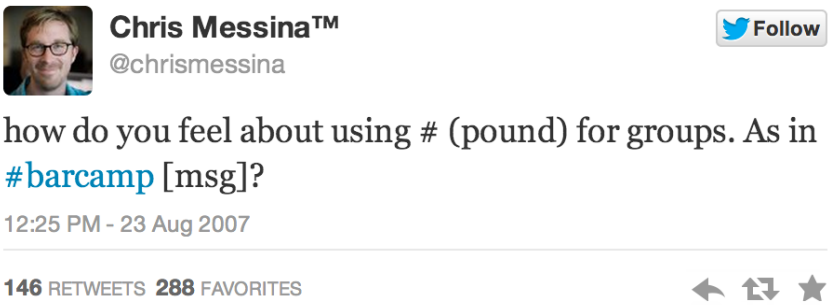


Figure 3.7: Example Hashtag Tweet

Beyond simply organizing your tweets, Twitter hashtags can help us craft your voice while joining in a larger discussion. We can use multiple hashtags in one tweet, but don't go overboard [10]. Many major brands now have Twitter accounts, and some choose to create hashtags to promote specific events or campaigns. A tweet that contains only hashtags is not only confusing it's boring. If your tweet simply reads, "#happy," your followers will have no idea what you're talking about. Similarly, if you tweet, "#BreakingBad is #awesome," you're not really adding much to the conversation.

Following

Following someone on Twitter means subscribing to their Tweets as a follower. Their updates will appear in your Home tab. That person is able to send you direct messages. Twitter works quite differently from social networks: when you accept friend requests on other social networks like Facebook, it usually means you appear in that person's network and they appear in yours. Following on Twitter is different because following is not mutual. Aggressive following is defined as indiscriminately following hundreds of accounts just to garner attention. However, following a few users if their accounts seem interesting is normal and is not considered aggressive. Aggressive follow churn is when an account repeatedly follows and then un-follows a large number of users. This may be done to get lots of people to notice them, to circumvent a Twitter limit, or to

change their follower-to-following ratio. These behaviors negatively impact the Twitter experience for other users, are common spam tactics, and may lead to account suspension [8]. In twitter every user can follow 2000 people total. Once you've followed 2000 users, there are limits to the number of additional users you can follow: this limit is different for every user and is based on your ratio of followers to following.

Followers

Twitter allows people to opt-in to (or opt-out of) receiving a person's updates without requiring a mutual relationship. People follow other users on Twitter to read updates that are interesting to them.According to the findings of [58], the motivation of users to follow a certain user is a shared topic interest between the follower and the followee. Additionally, the authors state that a followee following back a follower is also based on the fact that both users share the same interests. And 72.4% of all users follow back more than 80% of all of their followers (evaluated on a data set of 1,021,039 tweets posted by the top-1000 Singapore-based Twitter users). The authors showed that there is a homophily between followers and followees. The concept of homophily basically implies that people tend to be part of social networks in which the other members are similar to the user, e.g. in regards to attitudes, behavior, interests, education, sex, gender or race [51]. Additionally, Java et. el [42] identified three categories of Twitter users in regards to their following behavior: information sources, friends and information seekers. Information seekers are users who follow a large number of users, however only tweet rarely. A very similar categorization has also been observed in [44], where information sources are named broadcasters and friends are called acquaintances. However, as a third category, the authors add miscreants (evangelists) which are users who follow and contact a very high number of users aiming at getting attention and new followers.

Friends

Based on the categorization and definition by [42], friends are users who have a relatively similar number of followers and followees as they mainly are connected in reciprocal relationships. Our relationships can be measured by the attention we accord to people. We do so by interacting with them whether by making phone calls, meeting them for coffee, writing on their Facebook wall or in the case of Twitter, sending either direct or indirect replies. Interactions define the social relationship [39].

3.3 Feature Selection and Reasoning

The tweet text feature is considered to build the language model. The tweet text is used to compare the similarity of query tweet Q_t with the related set of tweets, which has the hashtags. Even though the text of the tweet is limited to 140 characters, the similarity measure helps to some extent. As the dataset captured from the user holds different languages which are also featured within the top-12 languages used on Twitter [60].The languages include English, Japanese, Spanish, Portuguese, Korean, French, Russian, Indonesian, Dutch, Turkish, Italian and German TweetSense considers only the English language to build the language model.

The temporal information of the tweet is used to find the tweets that are posted in a particular time window. This also helps to filter the hashtags, which are old and used less frequently. The hashtag, which is recently posted, is most likely to be the K most promising hashtags.

Trendiness helps to determine the hashtags that are frequently used by the users in his/her social network. This captures the insights of emerging topic in a user's timeline. Other than tweet related features, Twitter also emits social signals with in a social network of a user in terms of Retweets, Favorites, @replies and mentions and common usage of hashtags. And these set of features are categorized as user related features.

Retweets or Favorites provide the information on agreement and interest towards a particular user or topic. This is considered a social signal between users in a social network. Similarly, @replies and mentions provides the information on social interactions within Twitter. A user makes attention to another user through @replies and mentions. The number of people a user actually communicates with eventually stops increasing while the number of followees can continue to grow indefinitely [39]. The user may also share common hashtags with people who they are following. This indicates the likelihood between two users in terms of topic interests.

Other set of user related features include Following and Followers features are helpful to find the mutual friends and mutual followees to determine the most influential friend of a user. The HP's research [39] presents evidence to reciprocated attention. On average, 90 percent of a user's friends reciprocate attention by being friends of the user as well. Twitter users have a very small number of friends compared to the number of followers and followees they declare. This implies the existence of two different networks: a very dense one made up of followers and followees, and a sparser and simpler network of actual friends. The latter proves to be a more influential network in driving Twitter usage since users with many actual friends tend to post more updates than users with few actual friends. On the other hand, users with many followers or followees post updates more infrequently than those with few followers or followees.

A link between any two people does not necessarily imply an interaction between them. In the case of Twitter, most of the links declared within Twitter were meaningless from an interaction point of view. The Twitter interaction is not only based on the personal direct interaction but also influenced by the content shared by their followers. Therefore it cannot be counted as "meaningless from an interaction point of view". By combining the tweet content related and user related features that I discussed above, I will be able to predict the top K hashtags for a input query tweet.

Chapter 4

RANKING METHODS

In this chapter, I disucss the ranking methods of my system. For each canididate tweet in the tweet set of a input query tweet Q_t, I find a set of features based on its content and the user who posted the tweet. Based on these features,I use different ranking methods to compute the score for hashtags.The tweet content related feature includes tweet text, hashtag popularity and temporal information of the tweet. And the user related feature includes mutual friends, mutual followers and social signals like @mentions, favorites and co-occurrence of hashtags. I then combine all features scores to compute the rank for the candidate hashtag set to recommend the K most promising hashtag for a given query. Each candidate tweet in the set will have its own representation of feature scores, which are computed w.r.t to the features of an input query tweet. I describe the ranking methodologies in detail in the following sections.

4.1 Tweet Content Related Feature Scores

Similarity Score

This ranking method is based on the similarity between the tweet text of input query tweet Q_i and the entries contained in the set of tweets T_j with hashtag belonging to the social graph of the user who posted the query tweet Q_i. I assume the entries T_j that contain similar text contents will have more likelihood towards Q_i . So, this ranking method gives more weights to the hashtag belonging to the entries of T_j that have more word similarity to Q_i.

Before I choose the predominant function to compute similarity measure, I do initial preprocessing on the query tweet to extract the required information. As an initial step of the process, I only consider the tweets of English language and ignore query tweets of different languages. Considering the language style used in Twitter, stemming of words was not applied in this pre-processing step but I apply stop word removal using the $nltk$ [14] data. I also ignore the special characters and emoticons in the preprocessing step. Further, URLs and HTTP links are also ignored. After the final step of pre-processing the pre-processed input query is used to measure the similarity between the tweets. In TweetSense, I choose to use Cosine similarity on *TF-IDF* weighted vectors [43] to compute the similarity measure. Compared to other similarity measures such as Cosine similarity on BM25 Okapi weighted vectors [49, 54] , Dice coefficient [29], Jaccard coefficient [41] and Levenshtein distance [47], Cosine similarity on *TF-IDF* weighted vectors was found to perform well as stated by Eva et. al. based on experiments. [2].

$$\begin{aligned} \text{idf}\,(T_j) &= log\frac{N}{n(T_j)} \\ \text{tf-idf}\,(Q_i, T_j) &= tf(Q_i, T_j).idf(T_j) \\ \vec{Q_i}.\vec{T_j} &= \left\|\vec{Q_i}\right\| \left\|\vec{T_j}\right\| cos\Theta \\ cos\Theta &= \frac{\vec{Q_i}.\vec{T_j}}{\left\|\vec{Q_i}\right\| \left\|\vec{T_j}\right\|} \end{aligned} \tag{4.1}$$

In this case of searching for the best matching entry for a certain input query tweet,cosine similarity is computed between the query tweet and each and every tweet in the candidate tweet set. Let N be the candidate tweet set with hashtags and n_i be the number of tweets that contain keyword k_i of the query tweet. $F_q(i,j)$ is the raw frequency of k_i within tweet T_j. TF-IDF scores are computed to find Cosine Similarity between the Query tweet Q_i and candidate tweet T_j. The definition of Cosine Similarity is shown in

4.1. The final score computed based on the equation 4.1, that lies in the range of 0 to 1 is assigned to each Tweet and Hashtag pairs $<T_i, H_j>$. If more than one hashtag exist for a tweet, each one of the Tweet and Hashtag pair $<T_i, H_j>$ will receive the same similarity score as they are unique identity pairs.

Recency Score

Temporal feature of a hashtag is one of the most important features in ranking hashtags. Based on this ranking method, the hashtag that is temporally close enough to the query tweet is most likely to be used by a user. Especially in the case of dynamic microblogging platforms like twitter, the trending topics that keep evolving are based on how often a hashtag is recently used. In this ranking method, hashtags that are temporally closer to the query tweet gets higher ranking. I determine the time window of a Tweet, Hashtag pair $<T_i, H_j>$ by its "created at" timestamp $C(T_i)$. I adapt the exponential decay function to compute the recency rank of a hashtag. I use the following equation to compute the recency rank:

$$e^{\frac{C(T_Q)-C(T_i)}{60\times 10^k}} \tag{4.2}$$

Where k range from 0 to n, default $k = 3$

In the above equation, $C(T_Q)$ represents the created at timestamp of the query and $C(T_i)$ represents the created at timestamp of the candidate tweet. The time difference is converted to a representation of total seconds and divided by 60,000 seconds to compute the recency rank for a specify time window of 17hrs approx. In the above equation, k is used to set the time window w.r.t to the query tweet. Based on a small experiment on varying the sensitivity of the time window to 1 minute, 10 minutes, 2hours, 17hours and 170hours w.r.t to a query tweet whose ground truth hashtag is already known. The results are more promising when the time window is set to 17hours. Currently my system

default for k is 3, which is 17 hours. I then multiply value obtained from the exponential decay function to 10^4 as the values of the scores are so small interms of fractions. To make sure the final scores lie between lower bound 0 to upper bound 1, I do max normalization. I use max value to make sure the numbers are comparable to inputs.

Social Trend Score

The social trend rank computes the popularity of hashtags within the candidate hashtag set H_j of a particular query tweet Q_i and it varies for each Q_i. As the candidate hashtag set H_j is derived from the timeline of the user U_x who posted the query tweet Q_i, it is intuitive that the hashtag with high frequency is popular in the user's U_x social network. This can also be considered as tailored trending topics for a particular user. Social Trend rank is computed based on the"One person, One vote" approach. It is used to get the count of the frequently used hashtag in H_j. Only few hashtag are found to have high frequency based on its frequent usage. In this ranking method, the hashtag with high frequency is most likely a popular hashtag and will receive a higher ranking. To make sure to have the upper bound to 1 and lower bound to 0, I do max normalization as I did for the previous ranking method.

4.2 User Related Feature Scores

Attention Score

The attention rank captures the recent conversation between two users based on their @mentions and replies. This ranking method ranks the friends based on the recent conversations. If a particular user was @mentioned in recent times, it is more likely that they share topic of common interests. This also means they might use similar hashtags. Here the @mentions and replies acts as a social signal emitted between the users. This also helps to determine the tie-strength between the users. I compute user's attention rank

by a weighted average sum on the conversations between two users. Let $C(T_i)$ be the set of all tweets of user i and $C(T_j)$ be the set of all tweets of user j. And, $C(T_{i,j})$ be the set of all tweets which has @mentions and replies of i with j. $C(T_{j,i})$ be the set of all tweets which has @mentions and replies of j with i.Where j is the user who belongs to the list of friends of i. I compute the weighted average of @mention and replies between the users as below:

$$
\begin{aligned}
a_{i,j} &= \frac{\left|C(T_{i,j})\right|}{|C(T_i)|} \\
a_{j,i} &= \frac{\left|C(T_{j,i})\right|}{\left|C(T_j)\right|}
\end{aligned}
\tag{4.3}
$$

$$\text{Final Score } = (\alpha)\ a_{i,j} + (1-\alpha)\ a_{j,i}\ where\ \alpha = 0.5$$

The above equation gives equal importance to both user's attention. For example: Consider user i and j. If user i @mentions j and j doesn't reciprocal and vice versa. In this scenario the attention of one of the user is taken into account. If both users had conversations with each other, then both the users are given equal weights based on the α value. For my current system, I set $\alpha = 0.5$. This way social signals from both the users are equally estimated.

Favorite Score

The favorite rank is computed to rank the friends based on their favorites activity between the users. When user favorites a tweet posted by his/her friend, the user is directly letting his/her friend know that he/she is interested in that particular topic and shares agreement with him/her in that specific topic. This is a form of a social signal emitted by a user to another user and helps to determine the tie-strength between the users. Higher the number of times a user favorites a tweet of another user, he/she will

get higher ranking. I adapt the same computation method of weighted average sum that I used in the previous ranking method, to compute the favorite rank of the user.Let $C(T_i)$ be the set of all tweets of user i and $C(T_j)$ be the set of all tweets of user j. And, $C(T_{i,j})$ be the set of all tweets which has favorites of i with j. $C(T_{j,i})$ be the set of all tweets which has favorites of j with i.Where j is the user who belongs to the list of friends of i. I compute the favorite score between the users using the equation 4.3

Mutual Friends Score

Mutual friends are the people who are Twitter friends with both you and the person whose Timeline you're viewing. If a person i and j follow a same person X, then X is a mutual friend for both i and j. Mutual friends rank is computed to rank the friends based on their number of common friends they share in their social network. To find the similarity measure between two users based on their mutual friends relation, I modeled it as a set relation problem. Where one set F_i will contain the list of users followed by user i and the other set F_j will contain the list of users followed by user j. I use Jaccard's coefficient [41] to compute the mutual friends rank.

Mutual Followers Score

Mutual Followers are the people who are Twitter followers who follow both you and the person whose Timeline you're viewing. If a person i and j are followed by a same person X, then X is a mutual follower to both i and j. Mutual followers rank is computed to rank the friends based on their number of common followers they share in their network. To find the similarity measure between two users based on their mutual followers relation, I use the same computation method that I used in the previous ranking, as this is also a similar set relation problem. Here, one set FW_i will contain the list of users following user i and the other set FW_j will contain the list of users following user j. I use the same Jaccard's coefficient [41] to compute the mutual followers rank.

Common Hashtags Score

Common hashtags rank is computed to rank the friends based on the hashtags that are shared in common. If two users i and j use same set of hashtags for a particular time window, then both the users are talking about the same topic. So, more the number of common hashtags shared by a friend, higher the ranking he/she gets. To compute this, I first collect the unique set of hashtags used by each users. I then apply Jaccard's coefficient [41] to the unique hashtag set H_i and H_j of both the user i and his friend j. The final score is assigned to each of $i's$ friends. Based on this ranking method, each friend gets a unique scores by which they are ranked w.r.t to the user i.

Reciprocal Score

Reciprocal rank provides arbitrary weights for the users based on their following and following back terms. In Twitter, user's can follow anyone who they are interested. This also means that not all who the user follows are friends. The user might follow his friend but also follow a topic of his interest like news channel, celebrity etc., To give more importance to user's friends over others, the reciprocal rank assigns arbitrary values to differentiate the user's followers into friend and not a friend.For the users who follow back and forth each other will receive an arbitrary score of 1.0 and 0.5 other wise.

$$\left\{ \begin{array}{c} \varphi_m = 1.0 : \ if \ u \rightarrow v \ \& \ v \rightarrow u \\ \varphi_f = 0.5 : \ if \ u \rightarrow v \end{array} \right\} \tag{4.4}$$

Chapter 5

BINARY CLASSIFICATION

As discussed in the previous chapter, each candidate hashtags have its own set of features scores. In order to recommend the top K hashtags, all the feature scores of candidate hashtags have to be combined into a single final score. In this scenario, the scores are independent, continuous and subject to random variation. The range of all feature scores is normalized so that each feature contributes approximately to the final score. To approximate the final score we need a statistical model that models the previous observations. This problem definition falls under the inferential statistics, which are used to test hypotheses and make estimations based on sample data. Based on the inference and induction the system should produce reasonable answers when applied to well-defined situations and that it should be general enough to be applied across a range of situations. While each hashtags can be thought of as its own class, modeling the problem as a multiclass classification problem, poses certain challenges.Also, learning and classification will be a hard problem, as the average size of the class labels are in thousands. To overcome this challenge and to have a better prediction model, I modeled this problem as a binary classification problem.

In this case of binary classification problem, my training dataset will be a set of tweets with hashtags and my test dataset will be set of tweets without hashtags whose hashtag is to be predicted. I base my binary classification method by learning from a previous set of query tweets whose ground truth hashtag is known and try to predict the hashtags for the new set of query tweets whose hashtags are unknown. My training dataset contains the set of Tweet and Hashtag pair $< T_i, H_j >$ which are passed to my system to collect their candidate tweet set to compute their feature scores. Here the candidate tweet set

are the tweets from the timeline of the user U_x who posted $< T_i, H_j >$ pair. For each candidate tweet (CT), hashtag (CH) pair $< CT_i, CH_j >$ in the candidate tweet set, the feature scores are computed w.r.t to the $< T_i, H_j >$ as described in the Chapter 3. The final representation of my training dataset will be a feature matrix containing the features scores (predictor variables) of all $< CT_i, CH_j >$ pair belonging to each $< T_i, H_j >$ pair. Here all the feature scores are continuous except the class label is categorical. The class label for each $< CT_i, CH_j >$ pair will be a value of 0 or 1. The class label is 1 if the hashtag CH_j in the candidate set is equal to its hashtag H_j of its training tweet, hashtag pair $< T_i, H_j >$ and 0 otherwise. By this approach I transform the nominal classes (hashtags) to numerical classes.

Some of the functions suitable for learning binary classifiers include decision trees, Bayesian networks, support vector machines, neural networks, probit regression, and logistic regression. I started with choosing logistic regression and narrowed down deep intot he problem.As my contribution is my proposed method and not the learning algorithm, I use logistic regression as my predictor variables are continuous and my class labels are categorical. I use the prediction probabilities of the logistic regression model function to rank the candidate hashtags based on their probabilities. Also, Logistic regression deals with this problem by using a logarithmic transformation on the outcome variable, which allows modeling the non-linear association in a linear way.

5.1 Training

I collect my training dataset of Tweet and Hashtag pair $< T_i, H_j >$ from different set of users. I randomly pick the users through a partial random distribution by navigating through the trending topics in Twitter, along with a constraint that the user should follow at most 250 users. For each users, I take the most recent $< T_i, H_j >$ pair for training. My training dataset will include a $< T_i, H_j >$ pair which are scaled to its candidate tweet set.

Here, the candidate tweet set includes the tweets posted by the user and his/her friends. For each $< CT_i, CH_j >$ pair belonging to each $< T_i, H_j >$ pair, the feature scores are computed to form the feature matrix with the class labels as described before.Following Table5.1 represents a sample feature matrix

As described in the Table5.1, each row is a representation of a candidate $< CT_i, CH_j >$ pair represented by its feature scores and class labels. If the ground truth hashtag H_j of $< T_i, H_j >$ pair is present in the $< CT_i, CH_j >$ then the particular instance is a positive sample and if not the instance represents a negative sample. The instances with the class label 1 are defined as positive samples and the instances with class label 0 are defined as negative samples.

Attributes	Attributes Name	Value Range	Negative Sample	Positive Sample
λ_1	Similairty Score	0-1	0.149071	0.080162
λ_2	Recency Score	0-1	0.000029	0.992876
λ_3	Social Trend Score	0-1	0.670288	0.625012
λ_4	Attention Score	0-1	0.005621	0.001636
λ_5	Favorite Score	0-1	0.010263	0.014481
λ_6	Mutual Friends Score	0-1	0.011414	0.01016
λ_7	Mutual Followers Score	0-1	0.005376	0.006768
λ_8	Common Hashtags Score	0-1	0.023136	0.010274
λ_8	Reciprocal Score	0-1	0.5	1
λ_8	Class Label	1 or 0	0	1

Table 5.1: Table Representing the Training Dataset in the Form a Feature Matrix With its Class Label. Example for Positive and Negative Sample are Listed.

Since the occurrence of ground truth hashtag H_j in a candidate $< T_i, H_j >$ is very minimal, I face the problem of imbalanced training dataset due to higher number of negative samples and lesser number of positive samples for training. In multiple occurrences, my training dataset has a class distribution of 95% of negative samples and 5% of positive

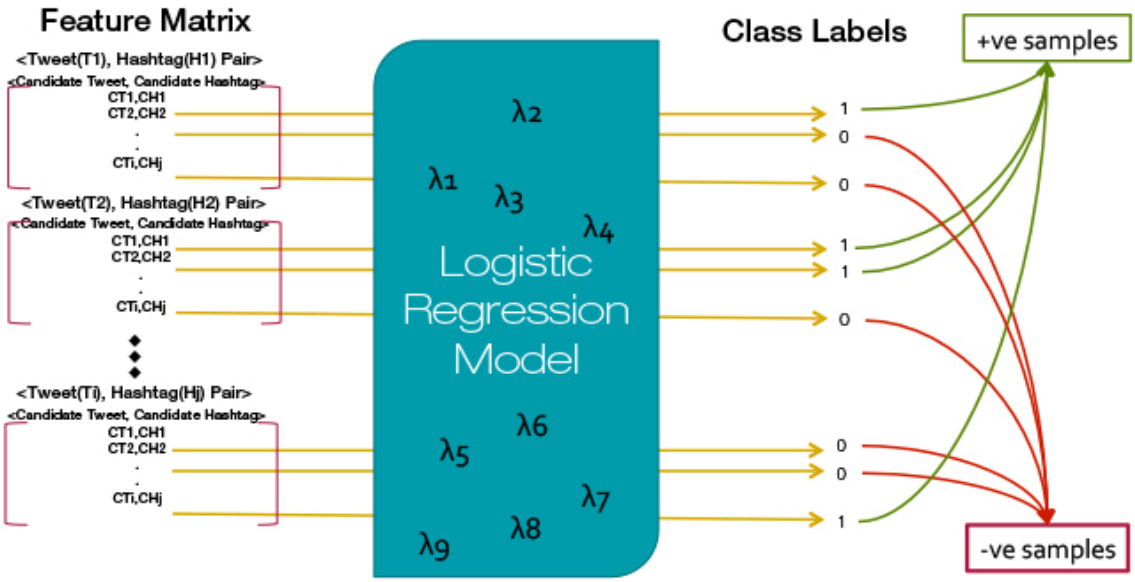


Figure 5.1: Training the Model from Tweet With Hashtags to Predict the Hashtags for Tweets Without Hashtag

samples. Learning the model from an imbalanced dataset will causes very low precision. To overcome this problem, under sampling and oversampling are proposed as the possible solutions. Based on the system by [25], I choose using Synthetic Minority Oversampling Technique (SMOTE) to resample the imbalanced dataset to a balanced dataset with 50% of positive samples and 50% of negative samples. As stated by Chawla et.al. in his paper [25], SMOTE technique does "an over-sampling approach in which the minority class is over-sampled by creating "synthetic" examples rather than by over-sampling with replacement. This approach is inspired by a technique that proved successful in handwritten character recognition [35]. SMOTE generates synthetic examples in a less application-specific manner, by operating in "feature space" rather than "data space". The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. In my case, I use the default five nearest neigh-

bors. For instance, if the amount of over-sampling needed is 200%, only two neighbors from the five nearest neighbors are chosen and one sample is generated in the direction of each. Synthetic samples are generated in the following way: Take the difference between the feature vector (sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features. This approach effectively forces the decision region of the minority class to become more general. The synthetic examples cause the classifier to create larger and less specific decision rather than smaller and more specific regions" [25]. Also that, the label is no longer guaranteed to be always predicted correctly.

Attributes	Attributes Name	Test Data Sample
λ_1	Similairty Score	0.092247
λ_2	Recency Score	0.083333
λ_3	Social Trend Score	0.470504
λ_4	Attention Score	0.470504
λ_5	Favorite Score	0.006098
λ_6	Mutual Friends Score	0.008621
λ_7	Mutual Followers Score	0.001887
λ_8	Common Hashtags Score	0.002621
λ_9	Reciprocal Score	0.5
Y	Class Label	?

Table 5.2: Test Dataset Representation

I then apply Randomization technique to shuffle the order of instances passed through the regression model to avoid bias and overfitting. The random number generator is reset with the seed value whenever a new set of instances is passed in. After applying the randomize filter the final balanced training dataset will have the instances in a random

order. I then, train my logistic regression model with this training dataset to learn the best fitting sigmoid function as shown in the Figure 5.1 to predict the test data.

5.2 Classification

Attributes	Attributes Name	Weights
λ_1	Similairty Score	-2.5535
λ_2	Recency Score	-6.17
λ_3	Social Trend Score	-6.0306
λ_4	Attention Score	-9.0566
λ_5	Favorite Score	-1.6062
λ_6	Mutual Friends Score	8.2656
λ_7	Mutual Followers Score	-3.192
λ_8	Common Hashtags Score	-38.4287
λ_9	Reciprocal Score	-0.4009
Y	Intercept	3.1536

Table 5.3: Logistic Regression Model Output

My test dataset includes the input query tweet Q_i that was passed into my system, whose hashtag is unknown. As discussed in the 5.1, my test dataset will be represented in the same format as my training dataset as a feature matrix with the class labels unknown. Following Table5.2 shows the sample feature matrix of the test dataset whose class label is unknown:

I then apply the logistic function, also referred as sigmoid function that was learnt from the training dataset, to predict the probabilities of the top k most promising hashtags. Logistic regression assumes that all data points share the same parameter vector with the query. Following Table5.3 shows the weights of each feature score that was computed using the logistic regression model.When I pass my test dataset to my logistic regression

Figure 5.2: Classification - Predicting the Probabilties for the Canididate Hashtags Belonging to the Input Query Tweet. If CH_1 is the Most Promising Hashtag for Query Tweet, It will be Labeled as 1 and 0 Otherwise.

model as shown in Figure 5.2, the logistic function predicts the maximum likelihood probability for each entry of candidate hashtag. If the predicted probability is greater than 0.5 then the model labels the hashtag as 1 and 0 otherwise. If a candidate hashtag is labeled as 1, it means that the particular hashtag is most likely to be suitable hashtag for Q_i. I recommend the top k promising hashtags for Q_i by ranking the hashtags that are labeled as 1 based on their probabilities. The hashtag that has the highest probability score of being predicted as class label 1 will get higher ranking in the list. In a similar way I recommend hashtag for all other input query tweets whose hashtags are unknown.

Chapter 6

EXPERIMENTAL SETUP

In the previous chapters, I focused on a specific approach of my system TweetSense, which involved recommending hashtags for a tweet based on their tweet content and user related features modeled through a logistic regression model. In order to prove my system is general enough to allow other variations for ranking hashtags, I describe a list of experimental setup that I will be using to evaluate my system. With the experimental setup, I describe some of the more compelling variations and discuss the relative trade-off with respect to TweetSense. In the next chapter, I will present the empirical comparison and evaluation of these variations to TweetSense.

6.1 Precision at N

I evaluate my proposed system by testing my system with a random set of tweets whose hashtags are deliberately removed for evaluation. So the correctness of my system is evaluated based on the presence of the deliberately removed hashtag in the top n recommended hashtags by my system. Based on this approach there will be only one relevant hashtag present in the candidate hashtag set. So the traditional information retrieval evaluation metrics such as precision and recall, do not directly account for evaluating the correctness of my system. In order to evaluate my system on relevance, I rank my system based on precision at n.So, If r relevant documents have been retrieved at rank n, then precision at n is r/n.The value of n can be chosen based on an assumption about how many documents the user will view. In Web search a results page typically contains ten results, so $n = 10$ is a natural choice. However, not all users will use the scrollbar and look at the full top ten list. In a typical setup the user may only see the first five results

before scrolling, suggesting Precision at 5 as a measure of the initial set seen by users. It is the document at rank 1 that gets most user attention, because this is the document that users view first, suggesting the use of Precision at 1 (which is equivalent to Success at 1).For example: Let's consider my system is tested with 100 tweets whose hashtags are deliberately removed for evaluation. Out of 100 tweets, if my system is able to predict the correct hashtag (in this case the hashtag which was removed from the tweet) for about 70 tweets at n= 5, then my precison at n=5 is 70/100. This means my system has 70% precision @ n. Also I consider the tweets with only one hashtag for the evaluation. Tweets with more than one hashtag and the retweets are ignored for evaluation. This is to make sure there is no bias involved in the evaluation metric. I perform this metric for different n values ranging from 5, 10, 15 and 20. The precision at each n value is plotted to determine performance of my system.

6.2 Precision at N on Varying Training Dataset

Currently my system is recommending the hashtags based on the model that I learn from a training dataset of tweet with hashtags. In order to test whether my system is general enough to recommend hashtags based on different cases of learning, I wanted to bring variations to my training dataset. I bring this variation by differing the size of the tweets, entries for those tweets and the number of users. By varying my training dataset I wanted to evaluate the precision @ n of my model for a fixed test dataset. In this experimental setup, I learn my model from a different set of tweet with hashtags ranging from $1-n$. I randomly pick the tweets from a set of users who are chosen from a partially random distribution. The number of users also varies from 1 to n. Each candidate tweet and hashtag pair in the training dataset will scale up to its candidate tweet set (timeline). The candidate tweet set size differs, based on the number of friends of the user and total number of tweets on their timeline. For example: For a partial randomly chosen set of

user $n = 5$, I pick a random set of tweets of size $n = 10$ for each users for which the candidate tweet set size differs based on their number of friends and numbers of tweets in their timeline. I vary the size of n for different cases to create several training dataset for learning. I then evaluate the accuracy of prediction of my model by using the evaluation metric that I defined in the previous Section6.1. I then determine if there is an increase in the precison at all value sof n from 5,10,15,20 for different training cases of the model. If my model performs better with respect to increase in the size of the training dataset, then my system is general enough to predict the hashtags based on learning.

6.3 Model Comparison with Receiver Operating Characteristic Curve

A receiver operating characteristic (ROC) curve is a graphical plot which shows the performance and accuracy of prediction of a binary classifier. In order to compare the binary classifier models learnt from different training datasets, I use ROC curve for prediction. ROC curve is plotted on the fraction of true positives (TPR = true positive rate) versus the false positives (FPR = false positive rate) out of the total actual negatives. The graph was plotted for different threshold settings. Here, TPR is called a sensitivity or recall curve. FPR can be calculated by 1- specificity. The notations are computed here:

$$Sensitivity = \sum \frac{TruePositive}{ConditionPositive} \tag{6.1}$$

$$Specificity = \sum \frac{TrueNegative}{ConditionNegative} \tag{6.2}$$

If the probability distributions for detection and false alarm are know, we can plot the cumulative distribution of the ROC curve from plus infinity to minus infinity. The ROC curve is plotted having false positives in x-axis and true positives in y-axis. The value of area under ROC curve measures accuracy of the classifier. The accuracy of the test depends how well the classifier classifies the true positives and false positives correctly.

As stated in in unmc article [4], area of 1 represents a perfect test; an area of .5 represents a worthless test. It says,

- .90-1 = excellent (A)
- .80-.90 = good (B)
- .70-.80 = fair (C)
- .60-.70 = poor (D)
- .50-.60 = fail (F)

As stated before, the area under ROC curve is equal to the probability that a classifier will rank randomly chosen positive instances higher than a negative instance.

6.4 Feature Scores Comparison Using Odds Ratio

As discussed in Chapter 4, I compute different features scores for my input tweet, based on its tweet content and user related features. I use these feature scores to form the feature matrix to learn their weights to combine their feature scores into a single final score to rank hashtags. In Tweet Sense, I am using nine independant feature scores to compute the probabilities of the recommended hashtags. Among the nine feature scores I wanted to find which feature score is contributing the most to the outcome of my final result set. In order to compare the feature scores, I use odds ratio [57]. As stated by Szumilas et. al in the paper [57],"Odds ratio is a well-known metric to measure the association between an exposure and an outcome. The Odds ratio represents the odds that an outcome will occur given a particular exposure, compared to the odds of the outcome occurring in the absence of that exposure. Odds mean the ratio of the probability of occurrence of an event to that of nonoccurrence. Odds ratio (OR, relative odds) is the ratio of two odds, the interpretation of the odds ratio may vary according to definition

of odds and the situation under discussion. The transformation from probability to odds is a monotonic transformation, meaning the odds increase as the probability increases or vice versa. The value of OR range from 0 to positive infinity. The OR is useful to determine the strength of association as: OR=1 Exposure does not affect odds of outcome, OR>1 Exposure associated with higher odds of outcome, OR<1 Exposure associated with lower odds of outcome. This helps to determine whether a particular exposure is a risk factor for a particular outcome, and to compare the magnitude of various risk factors for that outcome" [57].

As stated in UCLA FAQ [17], "by exponentiating the co-efficient, we get the odds ratios, which will say how much the odds increase multiplicatively with a one-unit change in the independent variable. In my scenario, all my co-efficients are continuous variables. For continuous variables the odds ratios can be interpreted as odds ratio between individuals who are identical on the other variables but differ by one unit on the variable of interest" [17]. For Example: the relation between the coefficient similarity rank λ_1 and its odds ratio is the logarithm of the odds of λ_1 over the odds of λ_1.

6.5 Ranking Quality

As stated by Herskovic et. al [37],"To compare algorithms, we need quantifiable evaluation metrics that account for ranking of retrieved results. Recall/precision curves are a traditional way of comparing entire retrieval process. Recall/precision curves indirectly reflect ranking and, as a result, may hide important differences between algorithms. Depending on the information need, users may be willing to review varying numbers of results. Choosing meaningful points in the retrieval pROCess may therefore be difficult. A natural extension of this idea comes from the field of statistical detection. Zhu suggests "hit curves", which graph cumulative detections versus number of cases pROCessed [61]. By analogy, the hit curves for a search pROCess graph cumulative "hits"

(relevant or important articles) versus position within the result set. Hit curves capture more information than traditional metrics and allow explicit evaluation of ranking. An additional benefit is the ability to restrict the result set to a manageable size while still being able to compare strategies in a standardized fashion. Further, hit curves capture what the user sees, rather than an abstraction. I adapt the existing evaluation metric Hit Curves [37] to determine relevance and importance of the ranked results by my system".

For a random set of tweet, I find the position of the relevant document in the result list. I take the number of counts at each K position of the result list to determine my systems ranking quality. My ranking quality will be determined higher if most of my recommended hahstag in the result list are in the top 4 ranking positions. For example: Lets consider the scenario of recommending top 10 hashtags for a random set of 100 tweets. If 60 out of 100 tweets have their correctly classified hashtag in the position 1 in the result list and 20 out of 100 tweets have their correctly hashtag at position 2 and so on. In this case, if the total number of times the correctly classified hashtag occurring in the ranking position 1 is higher compared to other ranking positions, then the ranking quality of the system is better

6.6 External Evaluation

In this section, I setup an experiment to evaluate my method TweetSense to an external baseline method proposed by Eva et.al [60]. Baseline approach on recommending hashtags is based on analyzing tweets similar to the tweet the user currently enters and deducing a set of hashtag recommendation candidates from these Twitter messages. They further presented different ranking techniques for these recommendation candidates. They propose four basic ranking methods for the recommendation of hashtags. These ranking methods are either based on the hashtags themselves (TimeRank, RecCountRank, PopularityRank) or the messages where the tweets are embedded in (Simi-

larityRank). SimRank ranking method is based on the similarity values of the input tweet input and the tweets containing the hashtag recommendation candidates CT. The cosine similarity has to be computed for every term within the input tweet and are used for the ranking of the recommendation candidates. TimeRank ranking method is considering the recency of the usage of the hashtag recommendation candidates. RecCountRank the recommended-count-rank is based on the popularity of hashtags within the hashtag recommendation candidate set. This basically means that the more similar messages contain a certain hashtag, the more suitable the hashtag might be. PopRank the popularity-rank is based on the global popularity of hashtags within the whole underlying data set. As only a few hashtags are used at a high frequency, it is likely that such a popular hashtag matches the tweet entered by the user. Therefore, ranking the overall most popular hashtags from within the candidate set higher is also a suitable approach for the ranking of hashtags. Beside these basic ranking algorithms, they propose to use several hybrid ranking methods which are based on the presented basic ranking algorithms. The combination of two ranking methods is computed by the following formula:

$$hybrid(r1, r2) = \alpha * r1 + (1 - \alpha) * r2 \tag{6.3}$$

where α is the weight coefficient determining the weight of the respective ranking within the hybrid rank. r1 and r2 are normalized to be in the range of $[0, 1]$ and can therefore be combined to a hybrid rank. Based on their evaluation, they state that the best results were achieved by combining the similarity of messages and the popularity of hashtags in the recommendation candidate set. In a benefit of doubt, I evaluate my system against their all three hybrid ranking method.

Chapter 7

EVALUATION AND DISCUSSION

In this chapter, I present an internal and external evaluation of my proposed approach TweetSense. I do this by comparing my system with the existing state-of -art system proposed by Eva et al. [60], as well as the variations I discussed in Chapter 5. I start by describing the dataset used for my experiments in Chapter 7.1. I then discuss my evaluation algorithm in Chapter 7.2, and then present results and discussions that demonstrate the merits of my approach in Chapter 7.3 and 7.4.

7.1 Dataset

The approach presented in this thesis and its evaluations are based on the underlying dataset of tweets, which is used to compute the hashtag recommendations. As there are no large Twitter data sets publicly available, I crawled tweets in order to build up such dataset. I use the official Twitter API for crawling. In general, a fraction of the twitter dataset is publicly available through the Twitter Streaming API (Application programming interface) enabling users to access it using certain methods. The Twitter Streaming API provides three levels of access to tweets namely the "Sprintzer" (which allows crawling upto 1% of all public tweets), "Gardenhose" (which allows 10% of public tweets) and "FireHose" (provides access to all public tweets). "Firehose" and "Gardenhose" are paid; the "Sprintzer" is our preferred method as it is freely available to the public. I use "Sprintzer" in my case. But crawling Twitter data through Twitter Streaming API has been constrained significantly by its rate limits. Rate limiting in version 1.1 of the API is primarily considered on a per-user basis or more accurately described, per access token. If a method allows for 15 requests per rate limit window, then it allows making 15 requests

per window per leveraged access token. Rate limits in version 1.1 of the API are divided into 15-minute intervals. In order to crawl a user's timeline, the method can only return up to 3,200 of a user's most recent Tweets. Crawling friends and followers id for a user has a limit of 5000 user ids. Also the favorite tweets that can be crawled are limited to recent 200 tweets per user.

With the above constraints, I crawl my tweets based on a user's social graph as my proposed approach is based on the generative model of exploiting the timeline of a user. I start with choosing a random set of users from a partial random distribution of users who are involved in the trending hashtags. I randomly pick the users by navigating through the trending hashtags on a particular time interval. For each partial randomly selected user, I crawl recent 1500 tweets and further crawl recent 1500 tweets for that particular user's friends who he/she is following. Since the tweets crawled to build a user's social graph is directly proportional to the number of friends. I randomly choose the user with a specific constraint that the user should have at most 300 friends. This helps to avoid overhead in computations. Other than friend's information, I also crawl the information about their followers who are following the user. If the user have more than 5000 followers then not all followers information are captured due to constraints with Twitter API of getting only 5000 follower ids for a user. By following this approach, I crawled 7,945,253 million tweets for a randomly picked user set of size n=63. Further details about the characteristics of the dataset can be found in Table7.1. The hashtag distribution is shown in Figure 7.1

7.2 Evaluation Algorithm

The evaluation of the proposed approach is done via a Leave-one-Out-Test [40]. Such a Leave-one-Out-Test is a traditional evaluation method for the assessment of the quality of recommendations. Basically, such an experiment is based on partitioning the test

Characteristics	Value	Percentage
Total number of users	63	N/A
Total Tweets Crawled	7,945,253	100%
Tweets with Hashtags	1,883,086	23.70%
Tweets without Hashtags	6,062,167	76.30%
Tweets with exactly one Hashtag	1,322,237	16.64%
Tweets with at least one Hashtag	560,849	7.06%
Total number of Favorite Tweets	716,738	9.02%
Total number of tweets with user @mentions	4,658,659	58.63%
Total number of tweets with Retweets	1,375,194	17.31%

Table 7.1: Characteristics About the Dataset Used for the Experiment

data (in my case the dataset mentioned in section 7.1). In the case of Leave-one-Out, one single item of the data set is withheld and constitutes the test item whereas the remaining entries are used for the computation of the recommendations. Subsequently, the recommendations are compared to the withheld item in order to evaluate the computed recommendations. My actual evaluation algorithm is described below:

- Divide the set of users for training and testing
- For each user in test set, randomly pick the tweet with hashtag and deliberately remove the hashtag for evaluation.
- Use this pre-processed tweet as an input to my system TweetSense.
- Run the system to get the recommended hashtag list.
- Verify if the ground truth hashtag exist in the recommended hahstag list.
- Compute the evaluation metrics such as precison at n and ranking quality as discussed in section 6.

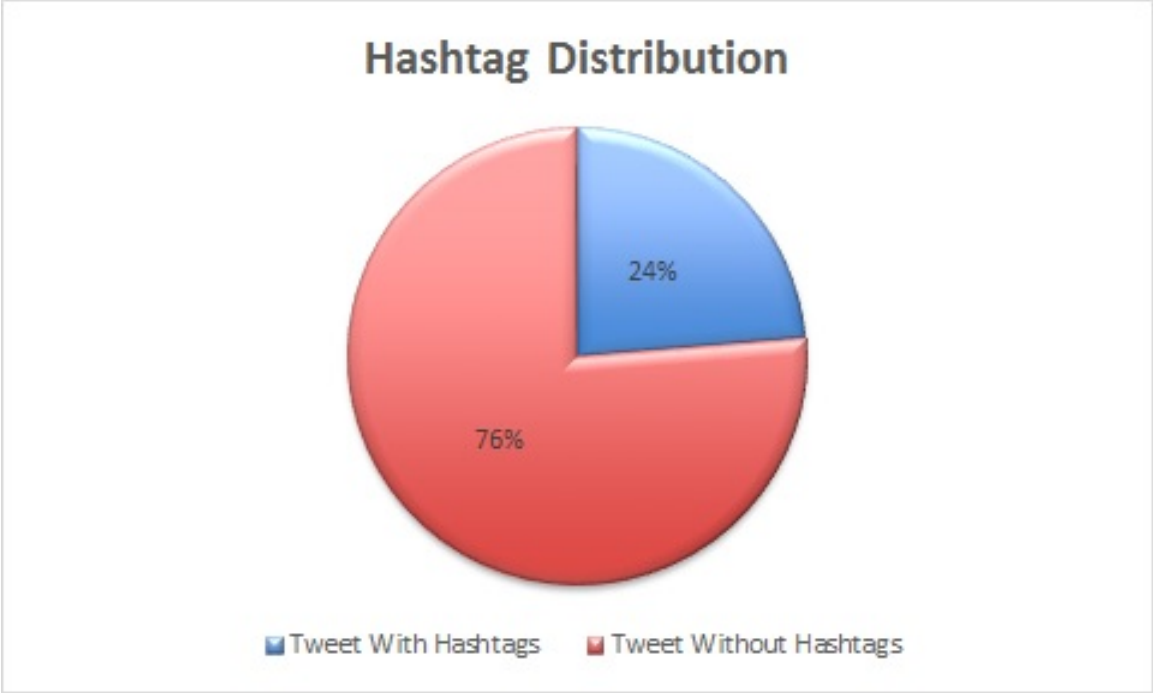


Figure 7.1: Hashtags Distribution

The evaluation was done on a RedHat EL6 machine with 8Gb RAM and one Quad-Core processor. I choose to use a sample size of 1599, i.e., for the evaluation, 1,599 random tweets are considered. I fix my test sample to 1599 tweets for all variation in evaluation process that I mentioned in section 6.

7.3 Internal Evaluation Of My Method

I evaluate my method based on the experimental setup that I mentioned in Chapter 6. In this evaluation, I assume my tweets are pre-indexed. I make this choice due to the constraints with Twitter API and limited access to Twitter firehouse. I index all my tweets in MongoDB database server for retrieval and computation.

7.3.1 Results of Internal Evaluation For Precision at N

I start with comparing the precision at n at 5,10,15 and 20 of the proposed system. This helps me to avoid the overhead of crawling the candidate tweet set from the user's social graph in real-time. I do the computation on the pre-indexed candidate tweet set to recommend hashtags. As I can see from the Figure 7.2, my approach gets better precision as the size of n is increased. For a total sample size of 1599 random tweets with hashtags whose hashtags are deliberately removed for evaluation. Total number of tweets for which the hashtag are correctly recommended by my system are represented in the graph in terms of percentage. At the value of $n = 5$, 720/1599 sample tweets are recommended with the correct hashtags. Similarly this range for 849/1599 at $n = 10$, 901/1599 at $n = 15$ and 944/1599 at $n = 20$. So there is a gradual increase of 15% in the retrieval accuracy from top 5 to top 20 ranging from 45% to 59% for this particular model. This retrieval accuracy should increase based on the increase in the size of the training dataset. I test this hypothesis with an experimental setup in next section.

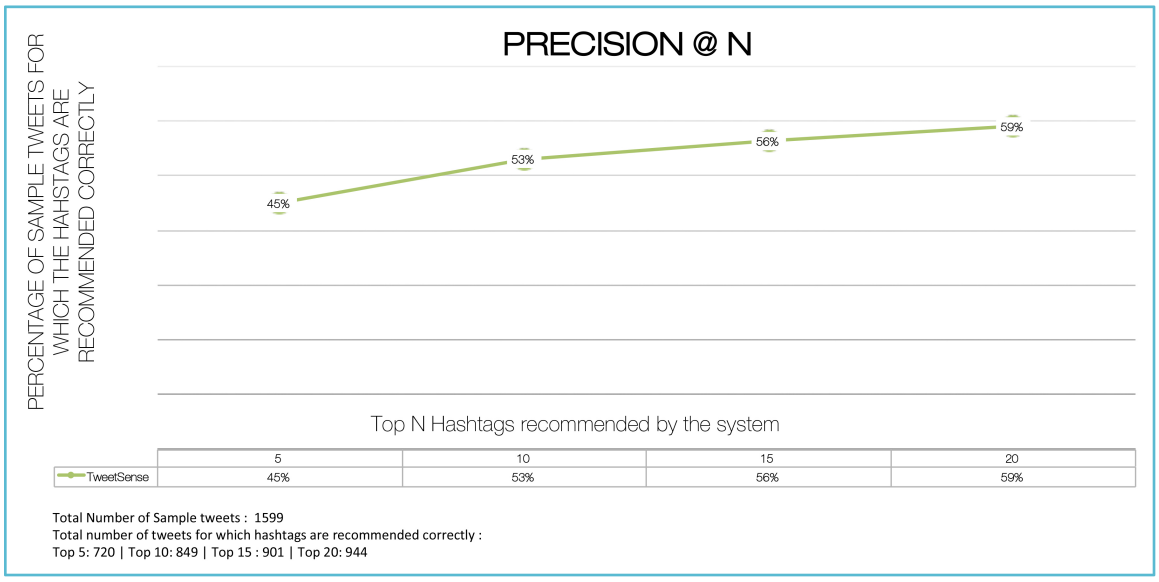


Figure 7.2: Precision at N for N =5,10,15 and 20 in Terms of Percentage.

7.3.2 *Results of Internal Evaluation Of Precision at N on Varying Training Dataset*

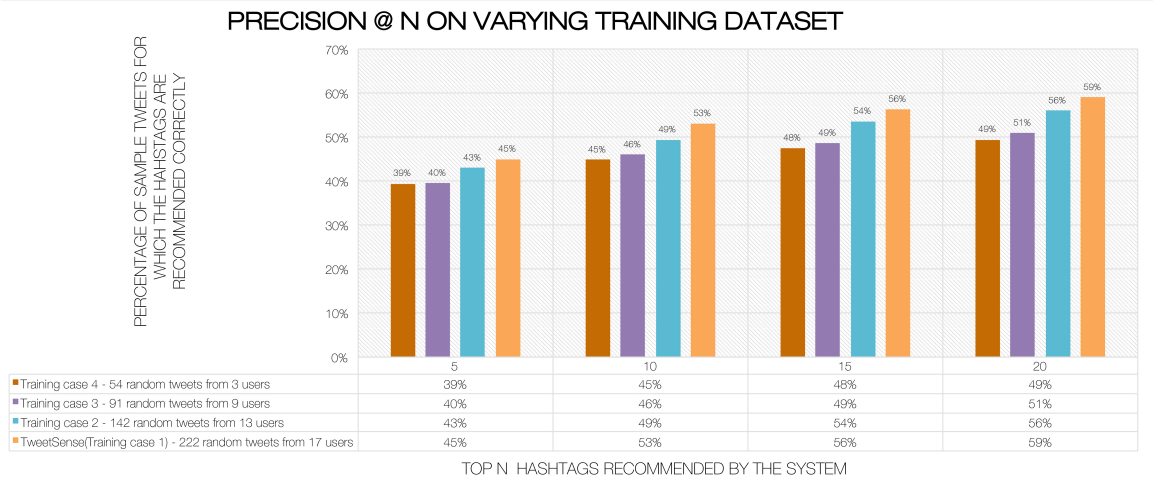


Figure 7.3: Precision at N = 5,10,15 and 20 on Varying the Size of the Training Dataset.

As mentioned in Chapter 6, I varied the size of the training dataset to measure the change in the precision value at each K value. I verify my hypothesis by computing the precision at n= 5,10,15 and 20. For this experimental setup, I have four different cases of training dataset for which I vary the size of the user and number of random tweets. Here, Training Dataset 1 has 222 random tweets with hashtags belonging to 17 random user, Training Dataset 2 has 142 random tweets from 13 users, Training Dataset 3 has 91 random tweets from 9 users and Training Dataset 4 has 54 random tweets from 3 users. The results are shown in Figure 7.1. Based on my approach discussed on Chapter 5, my random set of tweets scales up to their candidate tweet set. So at the phase of training, my Training Dataset Set 1 will have 10,568,685 instances, Training Dataset Set 2 will have 1,927,852 instances, Training Dataset Set 3 will have 695,064 instances and Training Dataset Set 4 will have 643,336 instances for learning. The results are shown in Figure 7.3

As expected, there is an increase in the precision value as I increase the size of my samples in training dataset. It also proves that the model learnt from a training dataset of higher sample size provides better accuracy. The comparison of my model learnt from

different training cases at each value of n shows an increase in the precision consistently. I currently use the model that I learnt from Training Dataset 1 as my system default.

7.3.3 Results Of Model Comparison with Receiver Operating Characteristic Curve

In addition to the model comparisons that I shown in the previous section, I also compare the performance of my model based on area under ROC curve as discussed in the chapter 7.3. I compare my model by fixing my test dataset. My test dataset in this case includes 70,585 instances. Of which model 1 correctly classified 66,979 instances, model 2 correctly classified 67235 instances, model 3 correctly classified 65006 instances and model 4 correctly classified 65002 instances.

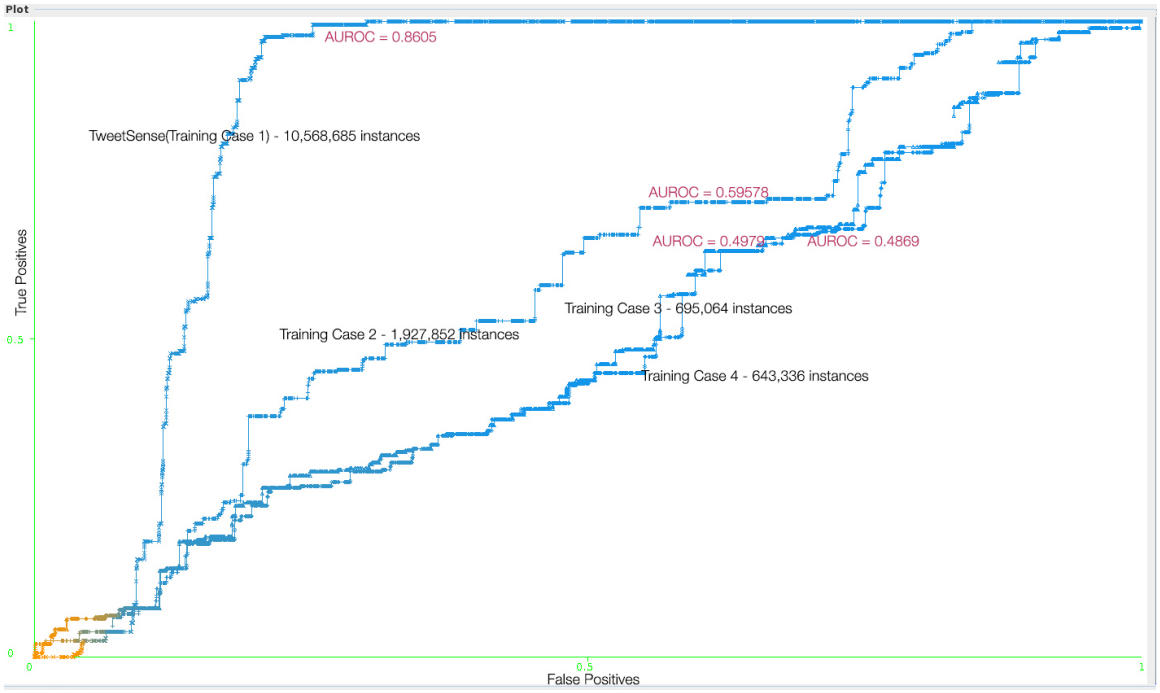


Figure 7.4: Model Comparison Based on Area under ROC Curve

As shown in Figure 7.4, the ROC curve for all four models shows that model 1 performs better than the other models with a higher value of Area under ROC = 0.8605. Comparatively model 2 has ROC value of 0.5957, model 3 has ROC value of 0.4979 and

model 4 has ROC value of 0.4869. Model 1 has a better area under ROC curve, as it was leant from higher number of instances over other models. The area under ROC curve of model 1 would be considered to be "good" at separating correct and incorrect hashtags for the input query tweet.

7.3.4 Results For Feature Scores Comparison Using Odds Ratio

My proposed approach on ranking hashtags is based on nine independent feature scores, which are based in the user related, and tweet content related features of the input query tweet. As my feature scores are used to compute the probabilities for the hashtag. I measure the association between the feature score and probability outcome using odds ratio. The odds ratio from TweetSense is shown in Figure 7.5 . Based on the measure that if the OR < 1, the variables are associated with lower odds of outcome and if OR >1, the variables are associated with higher odds of outcome.

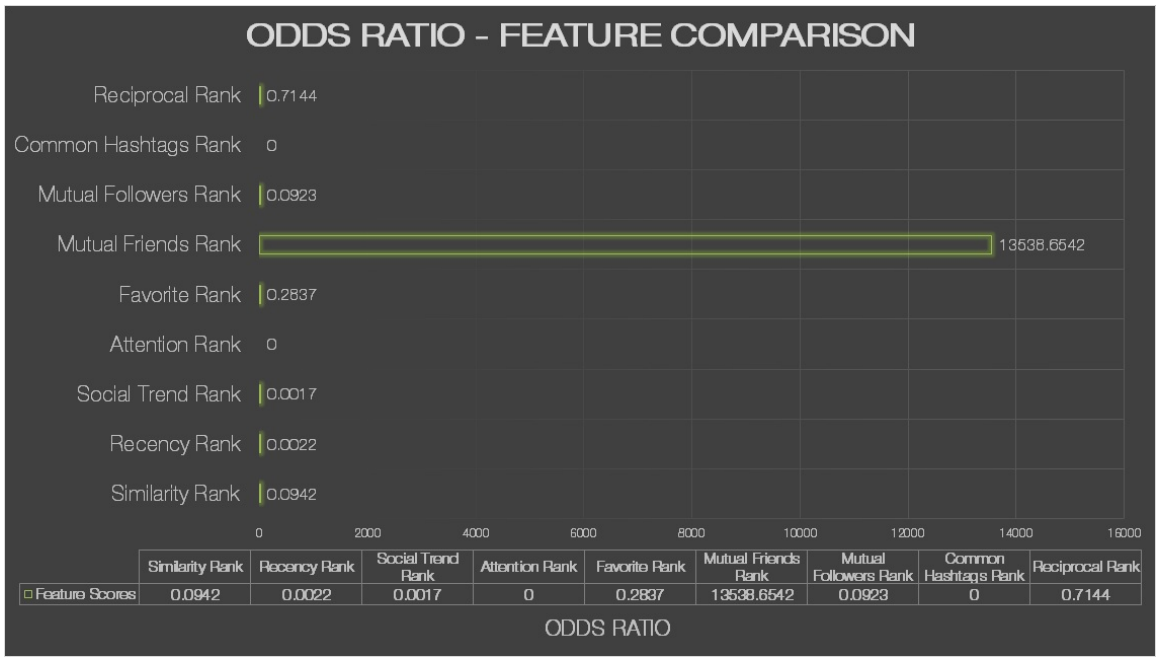


Figure 7.5: Odds Ratio for TweetSense with All Features

As show in the Figure 7.5, it seems only one feature "Mutual Friends Rank" is con-

tributing a lot to the odds of outcome compared to others. This also proves my hypothesis on giving importance to social signals over other tweet related features. In order to verify whether my model's prediction is based only on a single feature, I ran another experiment by ignoring the feature "Mutual Friends Rank". In this case as shown in the Figure 7.6, we can see that the higher odds of outcome goes to another social feature "Mutual Followers". This shows a correlation between the features. The reason could be due to feature redundancy.

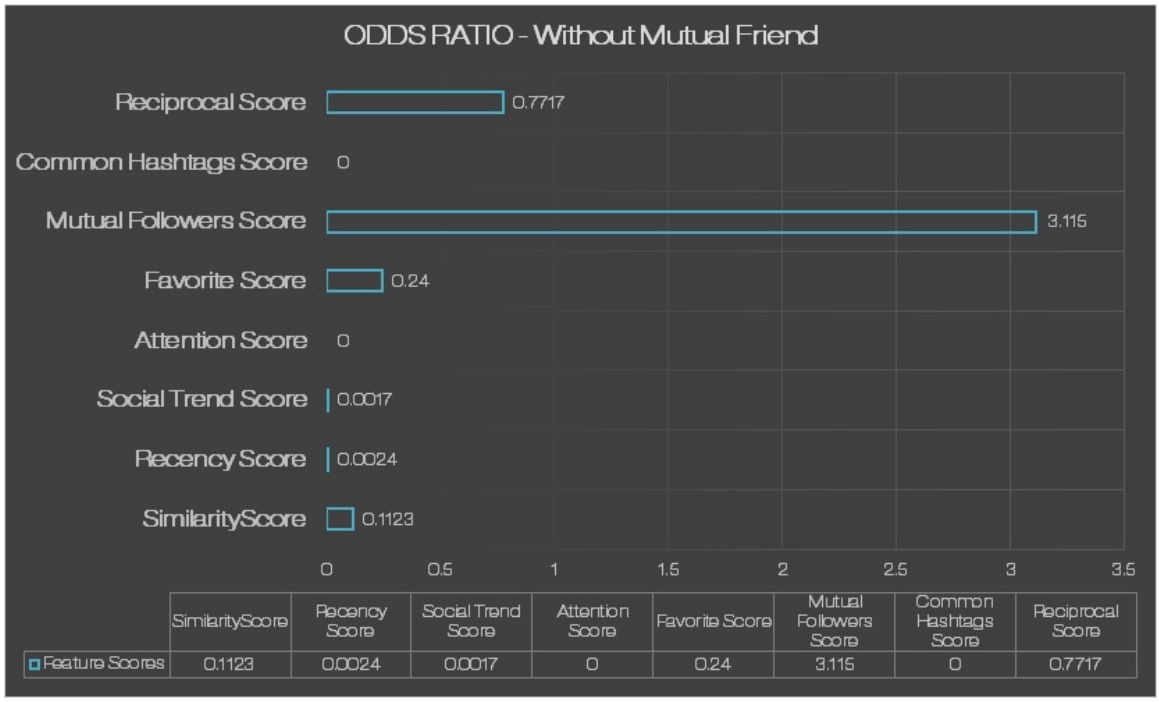


Figure 7.6: Odds Ratio - Feature Comparison - Without MutualFriend Score

In order to find how much the tweet related features affect the final odds of outcome, I eliminated the social features such as "Mutual Friend score", "Mutual Follower Score", "Reciprocal Score" to see how are the outcomes varied. As shown in the figure, Figure 7.7, we can see the outcomes are affected by the social activity for the user and the similarity score. All these experiments emphasize the fact that social features that I proposed are contributing more to he outcome and there is some correlation between features.

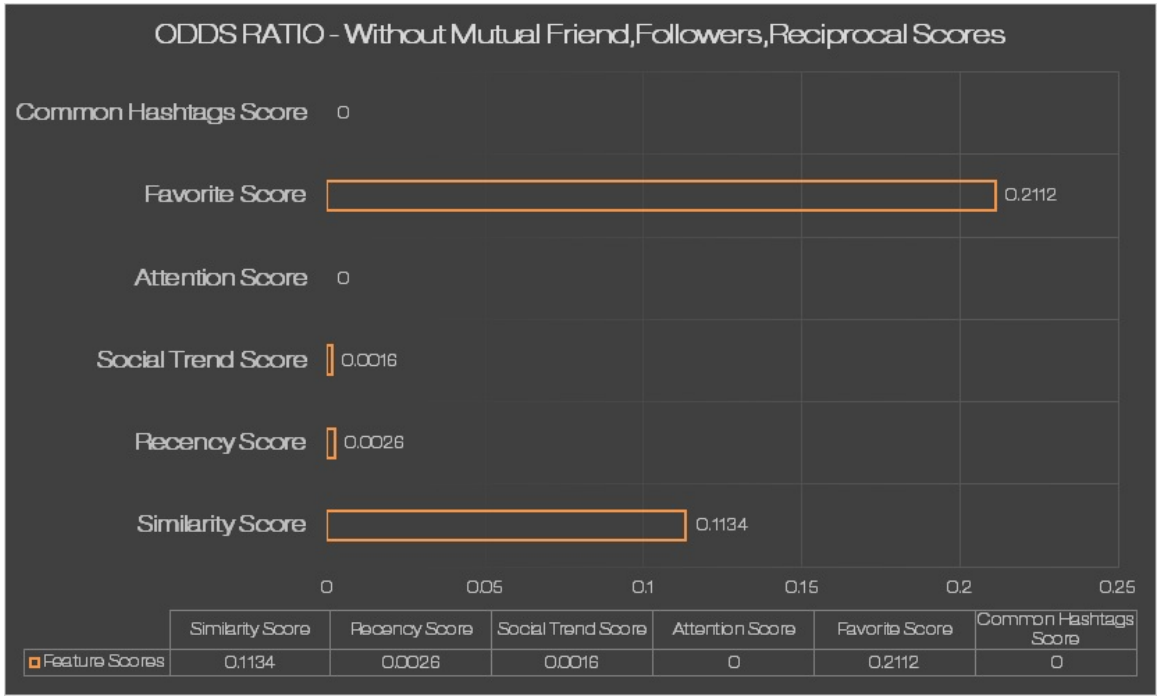


Figure 7.7: Odds Ratio - Feature Comparison - Without Mutual Friend, Mutual Followers, Reciprocal Score

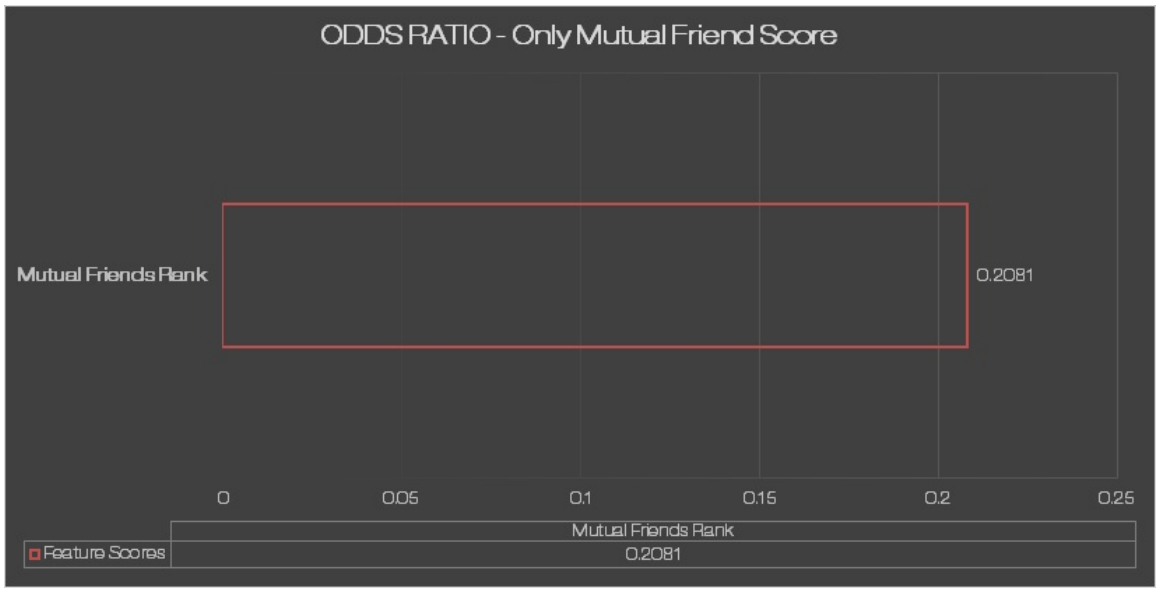


Figure 7.8: Odds Ratio - Feature Comparison - Only Mutual Friend Score

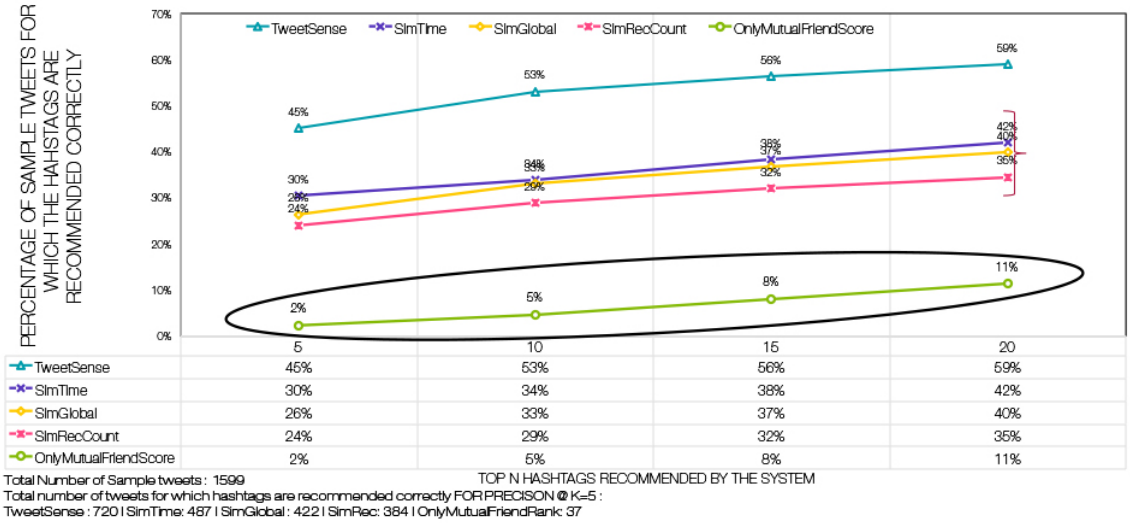


Figure 7.9: Feature Score Comparison on Precision @ N with Only Mutual Friend Score

I also ran another experiment to find the performance of my system based only on the single feature "Mutual Friend Score" as shown in Figure 7.8 that holds the lion share of the total outcome. As shown in the following figure Figure 7.9 we can see the performance of TweetSense is significantly low compared to my original system with all features and also with the base line system. This experiment also validates the fact that social features together with the tweet related features are contributing to the final outcome for my model.

7.3.5 Results Of Internal Evaluation Based on Ranking Quality

I evaluate the ranking quality for my system based on the position of the relevant hashtag in the result list as discussed in section 6.5. For the set of results discussed in Section 7.3.1, I scale the results in terms of ranking positions for the value of K at 1 to 10. As show in Figure 7.10, my system shows a higher-ranking accuracy by recommending the correct hashtags in top 4 positions. I also notice that most of the correctly recommended hashtags are in ranking position 1 and the next highest at ranking position 2 and so on.

So for a total sample tweets of n=849, 37.34% of the tweets are recommend with the correct hashtag at ranking position 1, 19.79% at ranking position 2 and the next highest 11.43% at ranking position 3 and so on. This consistent performance by my system on top 4 ranking positions proves that my system has a better ranking quality.

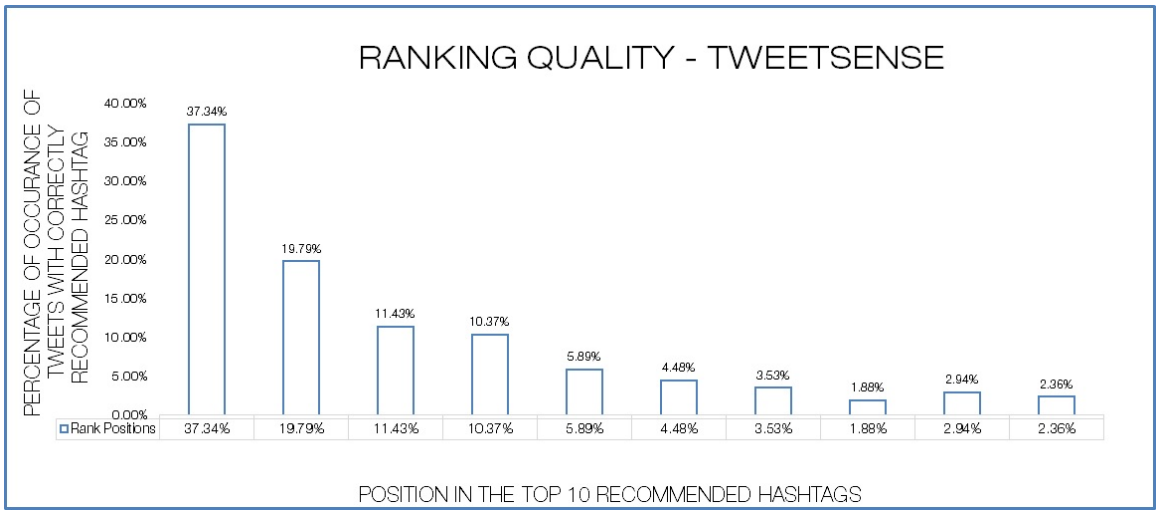


Figure 7.10: Ranking Quality for TweetSense

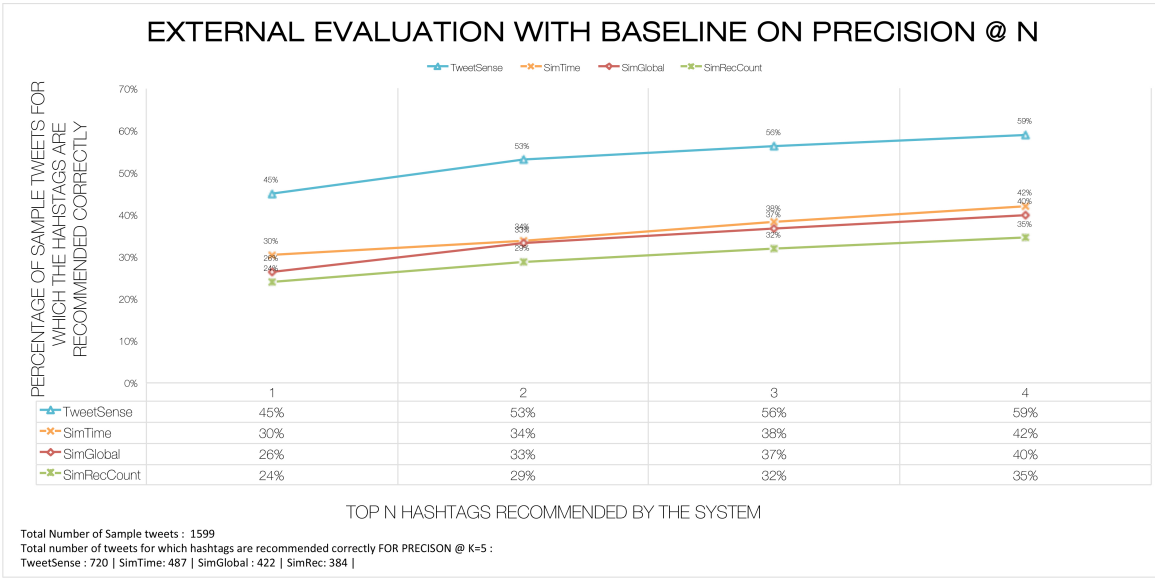
7.4 External Evaluation Of My Method

In this section, I evaluate my method,TweetSense,to an external baseline method proposed by Eva et. al. In order to compare my method to the external baseline, I implemented their system to the best of my understanding on referring to their published paper works and the code they shared with me which was written in Java. I make sure the algorithm that I implemented is exactly the same as described in their published paper. Although one particular hybrid ranking in the baseline method was mentioned performing better than others, to give them a benefit of doubt I compare my system with their all three hybrid-ranking methods described in their approach. I base my external evaluation based on the metrics of precision at N as I mentioned in Chapter 6. Further, the baseline system uses a different dataset for choosing their candidate hashtags. Since their

algorithm is independent of dataset, I evaluate their system on my dataset providing them the maximum potential to evaluate their system without any bias.

7.4.1 External Evaluation Of TweetSense Based On Precision at N

I compare the performance of my method over the baseline while assuming all the tweets are pre-indexed. As shown in Figure 7.11, my system TweetSense achieves more than 73% higher precision than the current state-of-art model proposed by Eva et. al. At the value of $n = 5$, my system was able to recommend correct hashtags for 45% of tweets compared to the best possible ranking method of the baseline model which could able to recommend only 30% of tweets with correct hashtags. On an average for different n values, my system dominates the baseline in all the cases with a significant increase in precision at n.



	1	2	3	4
TweetSense	45%	53%	56%	59%
SimTime	30%	34%	38%	42%
SimGlobal	26%	33%	37%	40%
SimRecCount	24%	29%	32%	35%

Figure 7.11: External Evaluation Againt State-Of-Art System for Precison @ N

7.5 Discussion

We have seen my system TweetSense achieves higher precision at n and ranking quality than the current state-of-art. In this section I hypothesize why my method works better than the baseline that I considered. I believe that recommending hashtags for a user should be not just be based on tweet related features but also the users previous history. A user tweets what he/she is interested and what he/she is exposed to in his/her timeline. Hardly a user would use a hashtag, which he/she has ever seen. For Example: A user living in UK is hardly to get exposed to a local TV show in Australia. So a better optimal way to recommend hashtag for that tweet is to look at the user's history and interests, rather looking into the global Twitter ecosystem. This way we make a better choice on selecting the candidate tweet set. Due to the dynamic nature of Twitter, I fall back to he belief that the most popular user in the social network is most likely to influence your status updates. Since I try to assess the most influential user in a user's social network, I use a statistical model that helps to determine the tie strength between the users. This model tries to validate the social signals such as mutual friends, followers, recent attentions, recent favorites etc., to determine the influential user. Also I believe that combining the tweet content related feature with user specific features would provide a better accuracy over the system that just considers the tweet content related features.

Let me consider a scenario to discuss how my method works in terms of recommending hashtags based on non-text similarity based feature and completely based on most influential users. For the following input query tweet "Los presents irrumpen en cantos de Gerald en apoya a", which is a non-english tweet whose ground truth hashtag is "#GoldenBoyLive", I compared the results of my system with baseline. In this case, my system was able to recommend the correct hashtag at rank position 2. Where as, all three hybrid-ranking methods of baseline method were failed to recommend hashtag as they

are highly dependent on the text-similarity based feature.

Further based on my experimental results, I notice that the hashtags belonging to the tweet with non-text similarity is recommended in the list of top 10 hashtags, which doesn't happen with the baseline model, which is completely based on the textual similarity and trendiness. I also notice my system recommends hashtags based on the most influential friend of the user who is active on a particular time span. I strongly believe that combining the features based on the social signals emitted by the user along with tweet content, recency and trendiness using a statistical model helped for the better performance of my system over the baseline. Also my ranking quality is significantly high, as most percentage of the recommended hahstags are in the top 4 positions in the result list.

Chapter 8

CONCLUSION

Twitter and other social networking sites are increasingly used to stay connected with the friends, tuned to their status updates, having daily chatters, conversations, information sharing and news reporting. The popularity and uncurated nature is susceptible to noise. With a huge social network in Twitter, information overload is a daunting factor that limits users social engagement and less active online users. But users in Twitter invented the peculiar conventions of using hashtag as a context or metadata tag for the tweet. Even though hashtags helps to derive the context for a tweet; they are not well adapted by Twitter users. So a biggest single factor that would persuade users to get engaged in their social network is to have a better recommendation system that recommends suitable hashtag for the tweets. This would inturn would increase the online social engagement. Twitter's utility grows as one's network grows; and one's network grows the more they interact with others. This in turn would also impact Twitter to maintain its monthly active online users in its social network.

As there is strong motivation and need for a better hashtag recommendation system, I propose a method to recommend hashtags based on the tweet content and user specific related features. I do this by starting with choosing the right candidate set of hashtags from the users timeline/ social graph based on the generative model of my system. I then extract the tweet content features such as test similarity, recency of the tweet and popularity and the social signal of the users such as mutual friends, mutual followers, recent favorites, recent direct replies, and follower-following relationship. I use a logistic regression model to union all the features that was extracted to recommend the most suitable hashtags. Based on the final prediction of my model, I rank the hashtags based

on their probabilities and recommend the top K most promising hashtags to the user.

My detailed experiments on a large twitter social graph constructed from the Twitter API shows that my proposed approach is performing better than the current state-of-art system proposed by Eva et. al. I also do internal evaluation of my system with various test cases to prove my model is general enough to learn and recommend hashtags for any input query tweet. Apart from the internal and external evaluation, I also show which feature score in my system contributing the most to the outcome. I also discuss the different training cases of learning my model using a partial random distribution of users. I also explain why I believe considering the user's timeline and history is a important approach for my system. And this intuition was later proved right using the empirical evaluations. I also try to explain why I believe TweetSense is performing better than other baselines and design choices considered. Thus, I propose a novel hashtag recommendation system for twitter users that considers the user's social signals and tweet content related features that achieves better results than the current state-of-art method on the same dataset.Furthermore, as users tend to use private and public lists and their influence towards a friends has more realtion towards the location, considering these features for recommendation are listed for future works. Taking location into account would make this system a location aware personalised hashtag recommendation system.

REFERENCES

[1] A closer look at twitter's user funnel issue,http://pull.db-gmresearch.com/cgi-bin/pull/docpull/1398-bdf2/23477782/0900b8c0880be965.pdf.

[2] Leveraging recommender systems for the creation and maintenance of structure within collaborative social media platforms,http://www.evazangerle.at/wp-content/papercite-data/pdf/evaphd.pdf.

[3] The first-ever hashtag, @-reply and retweet, as twitter users invented them,http://bit.ly/1kmjotl.

[4] The area under an roc curve,http://gim.unmc.edu/dxtests/roc3.htm.

[5] Replies vs. mentions on twitter,http://socialmedia.syr.edu/2013/04/replies-vs-mentions-on-twitter/.

[6] A field guide to twitter platform objects,https://dev.twitter.com/docs/platform-objects/tweets.

[7] Favoriting a tweet,https://support.twitter.com/articles/20169874-favoriting-a-tweet.

[8] Following rules and best practices,https://support.twitter.com/articles/68916-following-rules-and-best-practices.

[9] Using hashtags on twitter,https://support.twitter.com/articles/49309-using-hashtags-on-twitter, .

[10] The beginner's guide to the hashtag,http://mashable.com/2013/10/08/what-is-hashtag/, .

[11] Inside twitter's plan to fix itself ,http://qz.com/191981/inside-twitters-plan-to-fix-itself/?

[12] On twitter, what's the difference between a reply and a mention?,http://www.mediabistro.com/alltwitter/reply-mention_b34825, .

[13] What is the difference between @replies and @mentions?,http://www.hashtags.org/platforms/twitter/what-is-the-difference-between-replies-and-mentions/, .

[14] Nltk 3.0 documentation,http://www.nltk.org/api/nltk.corpus.html.

[15] Trending on twitter, http://www.hashtags.org/trending-on-twitter/.

[16] Twitter documentation,https://dev.twitter.com/docs/platform-objects/tweets.

[17] How to interpret odds ratios in logistic regression, http://www.ats.ucla.edu/stat/mult_pkg/faq/general/odds_ratio.htm.

[18] Signal over noise (with major business model implications),http://bit.ly/1kpcfyp.

[19] The problem with twitter by wahsington post, http://www.ats.ucla.edu/stat/mult_pkg/faq/general/odds_ratio.htm.

[20] What does favoriting a tweet mean?,http://bit.ly/q7lvvz.

[21] Meshary AlMeshary and Abdolreza Abhari. A recommendation system for twitter users in the same neighborhood. In *Proceedings of the 16th Communications & Networking Symposium*, page 1. Society for Computer Simulation International, 2013.

[22] Danah Boyd, Scott Golder, and Gilad Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10. IEEE, 2010.

[23] Hsia-Ching Chang. A new perspective on twitter hashtag use: Diffusion of innovation theory. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–4, 2010. ISSN 1550-8390. doi: 10.1002/meet.14504701295. URL `http://dx.doi.org/10.1002/meet.14504701295`.

[24] Hsia-Ching Chang. A new perspective on twitter hashtag use: diffusion of innovation theory. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–4, 2010.

[25] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *arXiv preprint arXiv:1106.1813*, 2011.

[26] Chen Chen, Hongzhi Yin, Junjie Yao, and Bin Cui. Terec: a temporal recommender system over tweet stream. *Proceedings of the VLDB Endowment*, 6(12): 1254–1257, 2013.

[27] Jilin Chen, Rowan Nairn, Les Nelson, Michael Bernstein, and Ed Chi. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1185–1194. ACM, 2010.

[28] Evandro Cunha, Gabriel Magno, Virgilio Almeida, Marcos André Gonçalves, and Fabrício Benevenuto. A gender based study of tagging behavior in twitter. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 323–324. ACM, 2012.

[29] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[30] Zhuoye Ding, Xipeng Qiu, Qi Zhang, and Xuanjing Huang. Learning topical translation model for microblog hashtag suggestion. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2078–2084. AAAI Press, 2013.

[31] Kate Ehrlich and N Sadat Shami. Microblogging inside and outside the workplace. In *ICWSM*, 2010.

[32] Wei Feng and Jianyong Wang. Learning to annotate tweets with crowd wisdom. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 57–58. International World Wide Web Conferences Steering Committee, 2013.

[33] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.

[34] Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 593–596. International World Wide Web Conferences Steering Committee, 2013.

[35] Thien M Ha and Horst Bunke. Off-line, handwritten numeral recognition by perturbation method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(5):535–539, 1997.

[36] John Hannon, Mike Bennett, and Barry Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 199–206. ACM, 2010.

[37] Jorge R Herskovic, M Sriram Iyengar, and Elmer V Bernstam. Using hit curves to compare search algorithm performance. *Journal of biomedical informatics*, 40(2): 93–99, 2007.

[38] Courtenay Honey and Susan C Herring. Beyond microblogging: Conversation and collaboration via twitter. In *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*, pages 1–10. IEEE, 2009.

[39] Bernardo A Huberman, Daniel M Romero, and Fang Wu. Social networks that matter: Twitter under the microscope. *arXiv preprint arXiv:0812.1045*, 2008.

[40] Albert Hung-Ren Ko, Paulo Rodrigo Cavalin, Robert Sabourin, and Alceu de Souza Britto. Leave-one-out-training and leave-one-out-testing hidden markov models for a handwritten numeral recognizer: the implications of a single classifier and multiple classifications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2168–2178, 2009.

[41] Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[42] 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. ISBN 978-1-59593-848-0. doi: 10.1145/1348549.1348556. URL `http://doi.acm.org/10.1145/1348549.1348556`.

[43] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[44] Balachander Krishnamurthy, Phillipa Gill, and Martin Arlitt. A few chirps about twitter. In *Proceedings of the first workshop on Online social networks*, pages 19–24. ACM, 2008.

[45] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[46] Su Mon Kywe, Tuan-Anh Hoang, Ee-Peng Lim, and Feida Zhu. On recommending hashtags in twitter networks. In *Social Informatics*, pages 337–350. Springer, 2012.

[47] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.

[48] Marek Lipczak and Evangelos Milios. Learning in efficient tag recommendation. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 167–174. ACM, 2010.

[49] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[50] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40. ACM, 2006.

[51] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[52] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, pages 385–388. ACM, 2009.

[53] Adam Rae, Börkur Sigurbjörnsson, and Roelof van Zwol. Improving tag recommendation using social networks. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*, pages 92–99. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2010.

[54] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *NIST SPECIAL PUBLICATION SP*, pages 109–109, 1995.

[55] Sandra Servia-Rodríguez, Rebeca P Díaz-Redondo, Ana Fernández-Vilas, Yolanda Blanco-Fernández, and José J Pazos-Arias. A tie strength based model to socially-enhance applications and its enabling implementation:< i> mysocialsphere</i>. *Expert Systems with Applications*, 41(5):2582–2594, 2014.

[56] Kaisong Song, Daling Wang, Shi Feng, Yifei Zhang, Wen Qu, and Ge Yu. Ctrof: A collaborative tweet ranking framework for online personalized recommendation. In *Advances in Knowledge Discovery and Data Mining*, pages 1–12. Springer, 2014.

[57] Magdalena Szumilas. Explaining odds ratios. *Journal of the Canadian Academy of Child and Adolescent Psychiatry*, 19(3):227, 2010.

[58] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270. ACM, 2010.

[59] Eva Zangerle, Wolfgang Gassler, and Gunther Specht. Using tag recommendations to homogenize folksonomies in microblogging environments. In Anwitaman Datta, Stuart Shulman, Baihua Zheng, Shou-De Lin, Aixin Sun, and Ee-Peng Lim, editors, *Eva2011*, volume 6984 of *Lecture Notes in Computer Science*, pages 113–126. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-24703-3. doi: 10.1007/978-3-642-24704-0_16. URL http://dx.doi.org/10.1007/978-3-642-24704-0_16.

[60] Eva Zangerle, Wolfgang Gassler, and Gunther Specht. On the impact of text similarity functions on hashtag recommendations in microblogging environments. *Eva2013*, 3(4):889–898, 2013. ISSN 1869-5450. doi: 10.1007/s13278-013-0108-x. URL http://dx.doi.org/10.1007/s13278-013-0108-x.

[61] Mu Zhu. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2004.