# A TIME-AWARE APPROACH TO IMPROVING AD-HOC INFORMATION RETRIEVAL FROM MICROBLOGS

ZAHRA AMIN NAYERI

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

MASTERS OF ARTS

GRADUATE PROGRAM IN INFORMATION SYSTEMS AND TECHNOLOGY
YORK UNIVERSITY
TORONTO, ONTARIO
APRIL 2014

# Abstract

There is an immense number of short-text documents produced as the result of microblogging. The content produced is growing as the number of microbloggers grows, and as active microbloggers continue to post millions of updates. The range of topics discussed is so vast, that microblogs provide an abundance of useful information. In this work, the problem of retrieving the most relevant information in microblogs is addressed. Interesting temporal patterns were found in the initial analysis of the study. Therefore the focus of the current work is to first exploit a temporal variable in order to see how effectively it can be used to predict the relevance of the tweets and, then, to include it in a retrieval weighting model along with other tweet-specific features. Generalized Linear Mixed-effect Models (GLMMs) are used to analyze the features and to propose two re-ranking models. These two models were developed through an exploratory process on a training set and then were evaluated on a test set.

# Acknowledgements

I would like to express the deepest appreciation to my supervisor Professor Jimmy Huang, who has shown the attitude and the substance of a genius and for his useful comments, remarks and engagement through the learning process of this master thesis. Without his supervision, support, and constant help this dissertation would not have materialized. I cannot thank him enough for his tremendous support and help. I feel motivated and encouraged every time I attend his meetings. Besides my advisor, I would also like to thank my committee members, professor Ali Asgary and professor Huaiping Zhu for serving as my committee members even at hardship. In addition, a thank you to Professor Georges Monette, whose passion for the data analysis guided me through the analysis section of the current work. I would also like to acknowledge with much appreciation the crucial role of Mahdis Azadbakhsh in helping with developing the re-ranking model. My sincere thanks also goes to Dr. Zheng Ye for accommodating me with his helpful advices throughout the course of the thesis. Last but not the least, I would also like to thank my parents for their endless love and support.

# Table of Contents

# List of Tables

# List of Figures

# 1 Introduction

Microblogging shares important characteristics with the more established practice of blogging such as the nature of the topics addressed and the nature of the affinities that brings audiences together. However they are different in terms of the text limitation. More precisely, users post short messages in the form of microblogs to express their ideas, feelings, interests, or to share news and updates. This information is particularly shared with the user's network of friends, but it can usually be accessible to the public depending on the users' privacy preferences. This is why a public stream of microblogs is a valuable source of information that can fulfill the needs of a variety of researchers, individuals, and business owners. This source of information is growing at a fast rate fueled by the growth in the number of microbloggers and their posts. Twitter is one of the most notable microblogging services that is very popular among users all around the world. In Twitter, users are allowed to share 140-character posts called ”Tweets”. What Twitter- similar to other microblogging platforms- introduces to the world is a network of people sharing information and thoughts. This platform raises lots of opportunities for research, since it has plenty of informative characteristics that can be used.

## 1.1 Motivations

As mentioned earlier, content production in Twitter occurs at a rapid pace. However, one might argue that the content is not worthwhile no matter how substantial in size. To investigate this matter we take a closer look at what different users post in Twitter. In a general categorization of users' activity in Twitter, Java *et al.* [20] suggest the following three activity classes, based on users' network connection, while excluding spam-tweeters:

- **Information seeking:** This refers to the activity of users whose behavior generally indicates curiosity in gaining information by following reliable sources in the area of their interest.
- **Information sharing, or the role of information source:** Activity of those users who use their own experience or other references to share information with their audience. News agencies, scientists, and reporters are examples of the members in this category.
- **Social Activity/The friend role:** The friend role is an inseparable part of the social networks. In most cases a small fraction of this type of user activity produces informative content. It is worth mentioning that the term ”informative” usually refers to whatever complies with one's interest; however, in this context it covers

a narrower range of interests. Precisely, the common chatter and phatic communication in Twitter have little value for many people. To the extent that in a report by Pear Analytics [23], these tweets that comprise 40.55 % of their sample of Twitter content, are referred to as ”pointless babble”. In practice, this category of tweets has shown to be valuable for a group of research purposes. These include work that applies sentiment analysis on Twitter for various reasons. As an example, Mitchell *et al.* [29] use this information to analyze levels of happiness and other factors in the United States and compares these measures by state. The research done by Sleeper *et al.* [42] is an example of another study that proves the usefulness of these types of tweets. They used crowdsourcing to find users who regretted posting tweets that criticize someone. This work replaced directly crawling tweets from Twitter’s public stream with crowdsourcing.

An even broader investigation on the characterization of the content of microblogs has been conducted on a data set crawled from Twitter in April 2009. This study is primarily focused on personal Twitter users and hence does not cover commercial and celebrity accounts. Accordingly, these types of content were not considered in this categorization [32].

- **Information sharing:** These are tweets sharing some piece of information. The information can be about politics, sports, news, and current events. Users share the information by announcing the event, sharing their thoughts about it, or discussing their ideas. Some tweets contain a URL with an article title and/or some commentary on its content. It is worth noting that for most retrieval applications, links to the web content can be considered as valuable sources of information.

- **Self promotion:** This type of activity occurs when Twitter users post about their recent work or publicize it. It is common for this type of tweet to have a link to a website.
- **Opinions/Complaints:** Tweets containing opinions or complaints about public figures and their actions, as well as different companies and their products. Tweets in this category are interesting for researchers in politics, social science and marketing, and are used to provide solutions for businesses. Jmal and Faiz [22] and Sommer *et al.* [43] used sentiment analysis on tweets to learn about customer reviews and improve customer relations respectively.
- **Statements and random thoughts:** Sometimes the tweeters share their perceptions and their everyday life events. In other words, they share their state of mind.
- **Me now:** What the users are doing and how they feel at the current state.
- **Question to followers:** Some tweets are aimed to ask the followers' opinions on a matter.
- **Presence maintenance:** Posting these tweets the user wants to announce her presence and activity in the timeline of her followers.
- **Anecdote (me):** Tweets sharing a story about the user with their followers.

- **Anecdote (others):** Tweets users post to share a story about others with their followers.

About 40% of the tweets were said to be in the ”Me Now” category. However, about 22% were ”Information Sharing”, around 25% are claimed to be ”Statement and Random Thoughts”, 24% fell into the ”Opinions/Complaints” category. These groups overlap and a tweet can belong to multiple categories. The numbers show that the content published in Twitter can satisfy the information need of various benefiters. About 20% of the messages could be considered more broadly relevant to a larger audience. With regards to the size of the content published by millions of users per second, this is a considerable amount of useful data. As mentioned earlier, microblogs are in the form of short-text. One of the widely noted properties of short texts is the sparse feature space that makes it difficult to discover correlations among the features. Immediacy and being nonstandard are among the other most important features of short text [55]. In addition to the content being brief in the case of microblogs, misspelling is common and non-standard language and structure is frequently used. So while there is some interesting content to be discovered on Twitter, it will definitely take work to be found. When we refer to the problems associated with the 140-character limit, we insinuate using textual features. However, there are different Twitter-specific features, both textual and non-textual that can be incorporated in the retrieval models when attending to the problem of scant features. For example, Twitter helps users specify the topics of their tweets using the # symbol (i.e.

Hashtag). There are other similar features worth investigating. Moreover, Twitter is a social network and there is a lot of information that can be derived from a network of this kind. Integrating these features into the retrieval model seems to be promising. No matter how intriguing, there is a tremendous amount of social network data and by including this information the problem gets more complex. The intent of this work is to use the above-mentioned features with a minimum added complexity. Moreover, we wish to come up with an efficient information retrieval framework to overcome the limitations introduced by sparse features in microblogs.

## 1.2 Contributions

The contributions of the proposed work are as follows:

To test the hypothesis that relevance can be predicted better using basic features, a number of algorithms were ran. The results suggested that a temporal pattern could be observed and used to predict relevance. Based on this, a time-based variable was proposed and integrated throughout the model. In order to enhance the temporal pattern and make it more consistent, different types of queries were investigated across multiple features. The result of this led to what we believe is the main contribution of this work: There is evidence that by using some customized temporal patterns it is possible to reach more accurate and consistent predictions of relevance.

We noticed that the basic features of the model have great untapped potentials to provide novel insights for our analysis. Based on this observation, we focused on the existing core features and avoided adding extra ones to the model. By performing statistical analysis on these features, we calculated the correlations between the features and the top ranked documents. We then used these correlations to estimate a better and a more accurate weighting model. The evaluation results prove the effectiveness and superiority of the proposed model.

A split-half method is used to achieve a valid estimate of the effectiveness of this model. This is done using a learning sample for developing the model and a testing sample to evaluate effectiveness and accuracy. The analysis was done against the gold standard of a data set that has been evaluated by human judgments of TREC experts.

Lastly, we used this temporal pattern-based approach to propose and validate a novel model for ranking tweets. This model uses a hybrid information retrieval approach and a modified BM25 for the purpose of microblog retrieval. It has also been tailored on the basis of our preliminary statistical analysis. As explained, the model reaches high consistency and accuracy in prediction of relevance manifested in the form of tweet rankings.

## 1.3 Thesis Structure

This dissertation is structured as follows: Chapter 2 focuses on the previous work and how the problem of microblog information retrieval has been addressed in the current literature; different perspectives have been considered for this purpose. Chapter 3 is dedicated to analysis of data before generating the model. This has been done in the form of a descriptive data analysis that focuses on the correlation of the features and tweet relevance. In chapter 4, the methodology and the proposed models are described. Two models for re-ranking are developed on the basis of logistic regression approach. These models are further elaborated in this chapter. The hybrid model and a modified BM25 have also been put forward. The experimental setting is presented and discussed in 5. A thorough explanation of the data set that is tailored for the ad-hoc retrieval task is also provided in this chapter. Chapter 6 is dedicated to the experimental results for both ranking and re-ranking models. This dissertation concludes in Chapter 7 with the inferred observations and exploration of possible future work.

# 2 Literature Review

## 2.1 Microblog Ad-hoc Retrieval

The focus of the current work is ad-hoc retrieval. In this section, different methods used for the microblog ad-hoc retrieval in the literature are reviewed. Kim *et al.* [24] identified three common themes in the literature: query expansion approaches; using temporal evidence to either re-rank the tweets or to incorporate it in the model by applying learning-to-rank [26]; and lastly custom functions. What follows is an overview of these three themes in the literature.

### 2.1.1 Query Expansion

Query expansion is identified as the first theme. It is very common to see inconsistency between the content of the tweet and a query since tweets are short. For example, for the query ”Detroit Auto Show” there can be a tweet referring to the same topic and using the word ”car” instead of ”auto”. Although the exact implementation of their methods differed, all of the top five runs in the TREC[1] 2011 conference included some form of query expansion [2, 13, 25, 28, 30]. With the exception of Louvan *et al.* [27], they all reported improved results after applying query expansion. Aboulnaga *et al.* [1] used ”Frequent Itemset Mining” [41] to expand the queries. The mined itemsets were built using the topics from the microblog track 2011. Terms were added first from the most relevant itemsets, then in another approach from clusters of itemsets. Although the results showed improvement on 2011 topics, they report no effect on the results for 2012 topics. Ibrahim *et al.* [19] used query expansion combined with two other techniques and reported improvement in their results. In their work, Zhu *et al.* [56] used query expansion along with learning-to-rank algorithms. They tested different methods of both query expansion and learning-to-rank against each other. Experiments of HIT group in TREC [14] was focused on the modeling of short text documents for combining text-based and non-text features, as opposed to applying learning-to-rank approaches. For this reason they examined the document expansion and query expansion in the classical language model framework. However, the results do not seem to be promising and they conclude that their document expansion approach is not a good solution.

[1] The Text REtrieval Conference or TREC is a continuing series of workshops co-sponsored by the National Institute of Standards and Technology (NIST) and the Intelligence Advanced Research Projects Activity. Different information retrieval fields have their own workshop and are referred to as ”tracks”. Starting 2011, TREC introdcued ”Microblog track” that is focused on information retrieval in tweets.

### 2.1.2 Temporal Distance/Relevance

Temporal distance or recency is also one of the popular themes. Some researchers used it as a feature in a learning-to-rank algorithm [28] and some others integrated it into re-ranking as a boosting factor [2, 13, 27, 39, 40]. Using this feature resulted in different outcomes. For example in the experiments carried out by Metzler *et al.* [28] the time feature got a 0 weight after training. Ferguson *et al.* [13] reported a negative impact for the experiments on the all relevant documents while positive impact was observed for highly relevant tweets [13]. The added temporal element resulted in minor improvements in the case of Roegiest *et al.* [40] and Amati *et al.* [2].

Rodriguez *et al.* [39] used a burst detection algorithm that extracts temporal features and reported improved results in both MAP [2] and P@30 [3] measures.

### 2.1.3 Custom Text-scoring Functions

With changing the parameters, some studies tweaked the scoring function and applied it to their microblog retrieval framework. Louvan *et al.* [27] changed the scoring function in Lucene in order to use binary term weighting. Ferguson *et al.* [13] also used binary term weighting and eliminated document length normalization by changing the

[2] For a set of queries, the mean of the average precision scores for each query is called Mean Average Precision (MAP).

[3] Precision in general is the fraction of the documents retrieved that are relevant to the user's information need. P@n is the precision measured at the given cut-off rank of n, considering the most relevant documents.

parameters in Okapi BM25 scoring function.

## 2.2 Use of Twitter-specific features in information retrieval purposes

Team COMMIT [8], PRIS [25], and IRSI [4] enlist a semantic expansion approach to deal with the problem of few existing features. They used external evidences such as Wikipedia or Google to generate new features. Petri *et al.* [35] used some keywords to generate a small number of the queries, while Obukhovskaya, *et al.* [33] and Amati *et al.* [2] used semantic expansion with internal evidence while taking advantage of pseudo relevance feedback method.

Other papers focused on hashtags and how to use them effectively in order to serve the purpose of adding more features. ISTI [5] proposes a method to split different words of each hashtag, while Bhattacharya *et al.* [6] suggests using the Tagdef service to find the most popular description of a hashtag.

The retweet[4] feature of Twitter helps users quickly share a tweet with all of their followers. Miyanishi *et al.* [30] and Zhang *et al.* [25] removed all tweets that begin with ”RT” or contain the HTTP status code of 302 that is the code indicating the retweeted posts. This was based on a claim that retweets are not relevant. Bandyopadhyay [4] also removed retweets that start with RT, but kept tweets that were of HTTP status code 302

[4] Repost or forward a message posted by another user. This action is often used for the purpose of spreading news or valuable information on Twitter.

and lacked the ”RT”.

URLs are very common on Twitter and are considered useful sources of information in tweet processing. Li *et al.* [25] removed URLs, while Amiri *et al.* [3] used URLs as spam detectors by excluding all tweets containing a URL for ”tinychat”, ”twittascope”, ”twitcam”, or ”twitcast”. Bhattacharya *et al.* [6] expanded short URLs and retrieve the Webpage’s title, keywords, and description with LongURL API.

Additional approaches to pre-processing were described in further papers. Efron [11] used a stopwords list of 133 terms; included in the list are some terms specific to Twitter (such as ”fb”, ”ff”, ”tinyurl”, and ”twitpic”). Li *et al.* [25] enlisted the traditional approach of stemming and transformation of text to lowercase. Similarly, Bandyopadhyay *et al.* [4] performed stopword removal and made use of the Porter stemming algorithm. Miyanishi *et al.* [30] used a clustering model to deal with Twitter TREC data set. Efron [11] excluded all tweets that contained more than four characters with encoding greater than 255 in order to identify and remove tweets that were not English. In contrast, Amiri *et al.* [3] discarded tweets that featured fewer than five words as they assumed they did not exhibit enough information. Tao *et al.* [45] proposed the use of named-entity recognition on the query strings and replaced letters occurring four or more times in sequence with a single letter. Finally, Wang *et al.* [47] suggested using a sample of 5,000 tweets instead of using all 16 million tweets included in the corpus.

Twitter-specific features also include both mentions [17] and retweets. These could be considered as different social network aspects of Twitter. Researchers take advantage of this characteristic in Twitter for the ad-hoc retrieval task in various ways. Cha *et al.*[9] conducted an experimental analysis of influence patterns in Twitter. The measures used were in-degree [5], retweets, and mentions. Based on this research, in-degree measure is independent from the number of retweets and mentions a user can get. Among the organizations and popular Twitterers, mainstream news organizations consistently get retweets. However, in the case of celebrities they mostly get mentions from their audience. Honeycutt and Herring [17] conduct a research that proves some users are using Twitter for informal collaborative purposes. Accordingly, the content in which users are mentioned cannot be neglected as conversation is an essential component of collaboration.

Moreover, Jiang and Scott [21] considered using the mentioning network[6] for the identification of information diffusion as opposed to network of followers. They believed that active users communicate and by this communication they create a hidden network. Moh *et al.* [31] used the information of followers and followings to distinguish spammers from legitimate users. The focus of their work is on each user's direct followers.

[5] The number of followers of a user, is his in-degree.

[6] See Appendix C

## 2.3 Using Probabilistic and Statistical Approaches

Qazvinian *et al.* [37] addressed the problem of rumor detection in microblogs. In their work, they built different Bayes classifiers as high-level features. Using these classifiers they learned a linear function for retrieval. They calculated the likelihood ratio to see whether they fall in the positive model or the negative model.

Wenyin *et al.* [49] studied text similarity in text-related research. The assumption was that it could be used as a measure to discover knowledge from textual databases. As opposed to long text, this work explored text similarity in short text that is associated with fewer features. The approach used both semantic information from WordNet and statistical information obtained from the corpus. While Wang *et al.* [48] proposed a more unified view of the retrieval task and divided the retrieval process into first predicting the relevance and then ranking the decision optimization stages.

The work of Chen and his team [10] consists of designing a scoring function with different retrieval goals and the final rank. They used an optimization approach and proposed a simple greedy algorithm. In their experiments, they considered various information retrieval metrics from the literature. They showed that with a relevance probabilistic model, the document could be directly ranked using the optimization result.

Zheng *et al.* [54] developed an algorithm based on the regression that can be applied to the objective functions involving preference data. Hence, the authors used regression as the basic element for development of a learning function. Miyanishi *et al.* [30] divided features into four scopes: ”Retrieval”, that gives search scores by different information retrieval models; ”Message”, which points to the features related to the tweet itself; ”User”, such as the number of followers; ”Semantic”, that indicates the conceptual difference between a query and a tweet. A two-step approach of consequently filtering and re-ranking was followed for the above four scopes.

# 3 Data analysis using generalized linear mixed model approach

The popularity of Twitter has made it a potentially reliable source for information; concurrently, the limitations of the nature of tweets have made retrieving the most meaningful information a complex procedure. Therefore many researchers have addressed tackling problems related to Twitter and proposed different approaches for ranking the relevant short text documents and yet the problem is unresolved. An essential step that is shared among these approaches is to decide what features to include in the retrieval model. In this chapter, we first introduce the data set together with the related definitions. Then a generalized linear mixed model approach to data analysis is presented. This analysis is conducted to identify the best features for inclusion in the model.

## 3.1 Data set

Twitter is a popular way of posting messages and updates among public figures, politicians, celebrities; companies and organizations along with a lot of normal people. However not everyone uses this medium for the same purpose. The purpose can even vary for an individual user from time to time. Topics of these tweets could be users' daily routines, feelings, or discussions and news on natural disasters, and events of global interest. Studies on the statistics of tweets reveal that there is a significant potential and a lot of information worth retrieving, but it is not easy to access that information in the public stream of Twitter in the first place. This is mainly a consequence of a rather recent change in the Twitter policy. Since then, the research community is facing the problem of limited access to API services. Hence, finding a unified reliable corpus for research purposes is a major problem. The Text REtrieval Conference (TREC), co-sponsored by the National Institute of Standards and Technology and U.S. Department of Defense added a microblog track in 2011. The data set is called Tweets 2011 and consists of around 16 million tweets from a two-week period that included the Egyptian Revolution and the U.S. Superbowl. Due to the earlier-mentioned limitations, this corpus is the most reliable one that currently exist. That is why this data set is used for the experiments in this work.

Tweets 2011 is associated with relevance judgment criteria in which, the tweets were judged on a three-point relevance scale and where human assessors read each tweet and judged its relevance to the associated topic queries. The data set is explained in further details in Chapter 5.

## 3.2 Temporal variable: day

As mentioned earlier, the approach taken in the current work is to capture the importance of the temporal feature in improving the ad-hoc retrieval task in the microblogs and to take advantage of it in the re-ranking model. Generally, temporal features can be identified in various ways. In this work, the variable ”day” represents the temporal feature and holds the value for the number of the days passed from a starting point. This starting point in the current setting is marked with the time of a ”trigger tweet”. A trigger tweet is defined by the assessors who contributed in shaping the corpus for the process of generating the queries. This process starts with the assessor searching the document, i.e. tweet, collection to get familiar with what it contains and the message it conveys. A search begins with an idea that that assessor has in mind. The assessor’s idea may have been triggered by something that was seen in earlier searches of the collection, or is just something the assessor is interested in. At some point the idea has jelled enough so that the assessor has an approximate idea of the number of relevant tweets in the collection. The aim is to have topics that have neither too many nor too few relevant tweets. The assessor formulates the query string and selects a tweet that acts as the trigger tweet. The trigger tweet is necessarily late relative to the set of relevant tweets because we need relevant tweets in the test set.

Table 3.1: Studied Features

| Feature | Level | Description |
|---|---|---|
| relevance | Tweet-Query | relevance scores, extracted from the qrel file |
| tweetLength | Tweet | number of characters in the tweet |
| hashTagCount | Tweet | number of hashtags in the tweet |
| URLExist | Tweet | whether there exists a URL or not |
| mentionCount | Tweet | number of mentions in the tweet |
| averageLength | Tweet | average length of terms of the tweet |
| numOccur | Tweet-Query | number of terms both in the query and tweet |
| numOccurHashTag | Tweet-Query | number of terms both in the query and tweet hashtags |
| timeDif.s. | Tweet-Query | difference between the tweet the query time (secs) |
| day | Tweet-Query | timeDif.s. in days |
| Query.Time | Query | time of the trigger tweet |
| Number.of.Terms | Query | number of terms in the query |
| Average.term.length | Query | average length of terms of the query |
| day | Tweet-Query | timeDif.s. in days |
| hashtag.rel | Tweet-Query | indicates presence of relevant hashtag; binary |
| hashtag.irrel | Tweet-Query | indicates presence of relevant hashtag; binary |
| hashtag.extra | Tweet-Query | number of add'l hashtags |

## 3.3 Descriptive analysis

In order to summarize the data and be able to better visualize it, some features are identified and their characteristics are studied. These features, or parameters, are extracted from an initial observation of microblogs and are in three different levels: tweet level, query level, and tweet-query level. Tweet level features are the ones that exclusively describe one aspect of a tweet. ”tweetLength” is an example of a tweet level feature, which holds the value for the number of characters in each tweet. On the other hand, if

Table 3.2: Summary of some of the features

| Variable | Min | 1st Quartile | Median | Mean | 3rd Quartile | Max |
|---|---|---|---|---|---|---|
| hashTagCount | 0 | 0 | 0 | 0.416 | 0 | 16 |
| mentionCount | 0 | 0 | 0 | 0.291 | 0 | 18 |
| numOccur | 0 | 0 | 0 | 0.664 | 1 | 15 |
| tweetLength | 4 | 74 | 109 | 101 | 136 | 140 |
| averageLength | 3.03 | 4.33 | 4.93 | 5.05 | 5.60 | 9.90 |

the variable is only taking the query attributes into account, it is categorized as a query level variable. Those variables that have both attributes from the tweet and from the query are tweet-query level. Table 3.1 gives an overview of the features providing a short description of the feature for further reference.

The dependent variable ”relevance” was transformed into a binary form by combining the relevant and highly relevant into a single category of relevant tweets, we mark spam tweets as ”N/A” and exclude them from the experiments, and treat the rest of the tweets as irrelevant. This was done for the purpose of using logistic regression and its related analysis that comes further in section 3.4.

Summary of some of these features are shown in Table 3.2. All of the figures in this table are extracted from the training subset of the data set. According to Table 3.2, maximum number of hashtags in a tweet is 16 in the current data set. A histogram in Figure 3.1 complements the picture by showing how rare it is for the tweets to have four or more hashtags, and most of them contain zero or very few hashtags. This is also true about the number of mentions in the tweets: while it can be as high as 18, as shown in Figure 3.2, it is in most cases less than 5. Even though lots of the tweets have neither hashtag nor mention, there is still a considerable number of them containing these features. It can be observed that occurrences above 5 are rare.

As noted in Table 3.1, ”numOccur” refers to the number of terms that are occurred both in the query and the tweet. Maximum number of occurrences was 15 according to Table 3.2. However the frequency of occurrences is similar to ”mentionCount” and ”hashtagCount”. The histogram for this is depicted in Figure 3.4 and shows that a significant number of tweets have at least one occurrence of similar terms in queries. This calls for further investigation and shows that ”numOccur” might contribute in determining the relatedness of a tweet to a query.

The next studied variable is ”tweetLength”, or the number of characters in each tweet posting. According to Table 3.2, the value of ”tweetLength” varies between 4 and 140, which is the maximum number of characters allowed by Twitter service. The mean is 101, and the third quartile is 136, implying that the limit bounds users to keep their posts as long as 140 characters and accordingly 25% of the posts have more than 136 characters. This can also be seen in Figure 3.4, as the frequency of tweets raises significantly towards the end of the range. While in the first three quartiles the number of characters is rising smoothly. Very few tweets have minimal character counts.

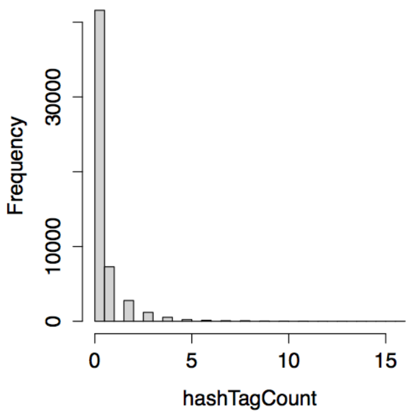


Figure 3.1: The frequency of different hashtag counts among tweets corpus

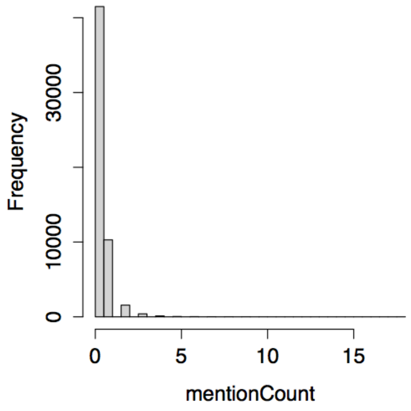


Figure 3.2: A histogram showing how the number of mentions varies in tweets

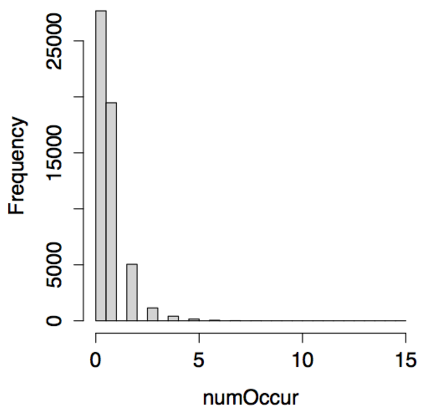


Figure 3.3: The histogram of similar term occurrence in tweets and queries

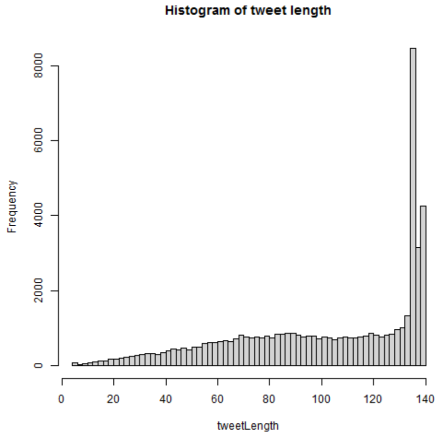


Figure 3.4: The number of characters, i.e. tweet length histogram

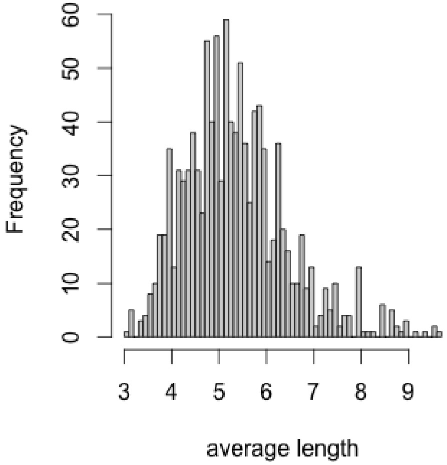


Figure 3.5: The average length of tweets

The variable ”averageLength” shows the average length of terms in the tweets. Range of this variable is between 3.03 and 9.90. Nevertheless, 50% of the data is between 4.33 and 5.60 and is in accordance with average length of English words, which is about 5.1 letters [7]. This measure can shed light on the content of a tweet implicitly. Based on a previous research, the average intelligibility of words increases with their length. However we cannot expect large averages in length because of the low frequency of the long words [18]. Figure 3.5 depicts the histogram of average length and shows how most of the tweets follow the average word length of words in the English language.

Lastly, the number of similar terms in tweets and queries with regards to the time variable (day) is studied. In order to do so, the queries are divided into two levels: queries with higher ratios of relevant tweets, Figure 3.6, and queries with lower ratio of relevant tweets in their corresponding tweet subset 3.7. Each panel represents a query and the percentages in parentheses next to the query ids show the ratio of relevant tweets over all of the studied tweets for each query. In most cases, the probability of similar terms to occur in the tweets remains static as the time passes. However the pattern is not consistent for all of the queries. For the queries number 68 and 10, a dramatic fall of the number of similar terms is noticeable. The fall is less sharp in the case of queries 61, 47, and 48.

[7] http://www.wolframalpha.com/input/?i=average+english+word+length

## 3.4 Time-aware analysis of features

Since the focus of this work is on the time-aware aspects of Twitter information retrieval, the temporal variable introduced earlier in 3.2 is further studied in the following section. The aim is to see if there exists a pattern of relevance in the tweets with regards to time. Figure 3.8 depicts how the relevance of tweets varies over days. This graph suggests that relevance has a very strong relationship with time. This provides the basis for further exploitation of this relationship at the query level. In addition, it can demonstrate that tweets with URLs tend to have more probability to be relevant and as the number of similar terms in the tweet and query rises the relevance improves.

Time of the trigger tweet is set to be the starting point of analysis. Two peaks can be seen in the curve for this range. Given the nature of Twitter, most of the topics are news-like searches. News in general and Twitter in particular, tend to be bursty. In other words, there are a lot of posts on the event shortly after it occurs and then the interest in the event wanes.

Figure 3.9 studies the temporal variable against relevance separately for different queries in the training set. One can simply note the dramatic variability between queries when studying relevance with regards to time. The red lines represent the estimated proportion of relevant tweets based on the scores in the qrel files and as a function of time. The qrel files include the human judgments for the relevance of the tweets and are usually used for evaluation of retrieval as a gold standard. This figure uses stat_smooth [50] for the estimations. The panels are ordered according to the mean relevance for each query. Looking at these different panels, we can see that the patterns are highly variable. For some queries, recency is very relevant in determining relevance and we would have some patterns where it is not. Query 11 is an example of a pattern with a highlighted importance of recency, while in case of query 8, there is a resurgence of interest a couple of weeks after the posting time of the trigger tweet.

To adjust the temporal function, ranking scores from a baseline model that uses BM25 were used. This was done to investigate whether using a conventional ranking function can improve the consistency of the pattern among different queries. An adapted partial residual plot, where y axis was only adjusted against BM25, is depicted and shown in Figure 3.10. Even though there is a slight improvement in consistency of the patterns, there seem to be more factors affecting the relevance. These findings lay a path for future research since they show that it is important to identify what characteristics of queries are related to time patterns; in order to get access to an additional source of information.

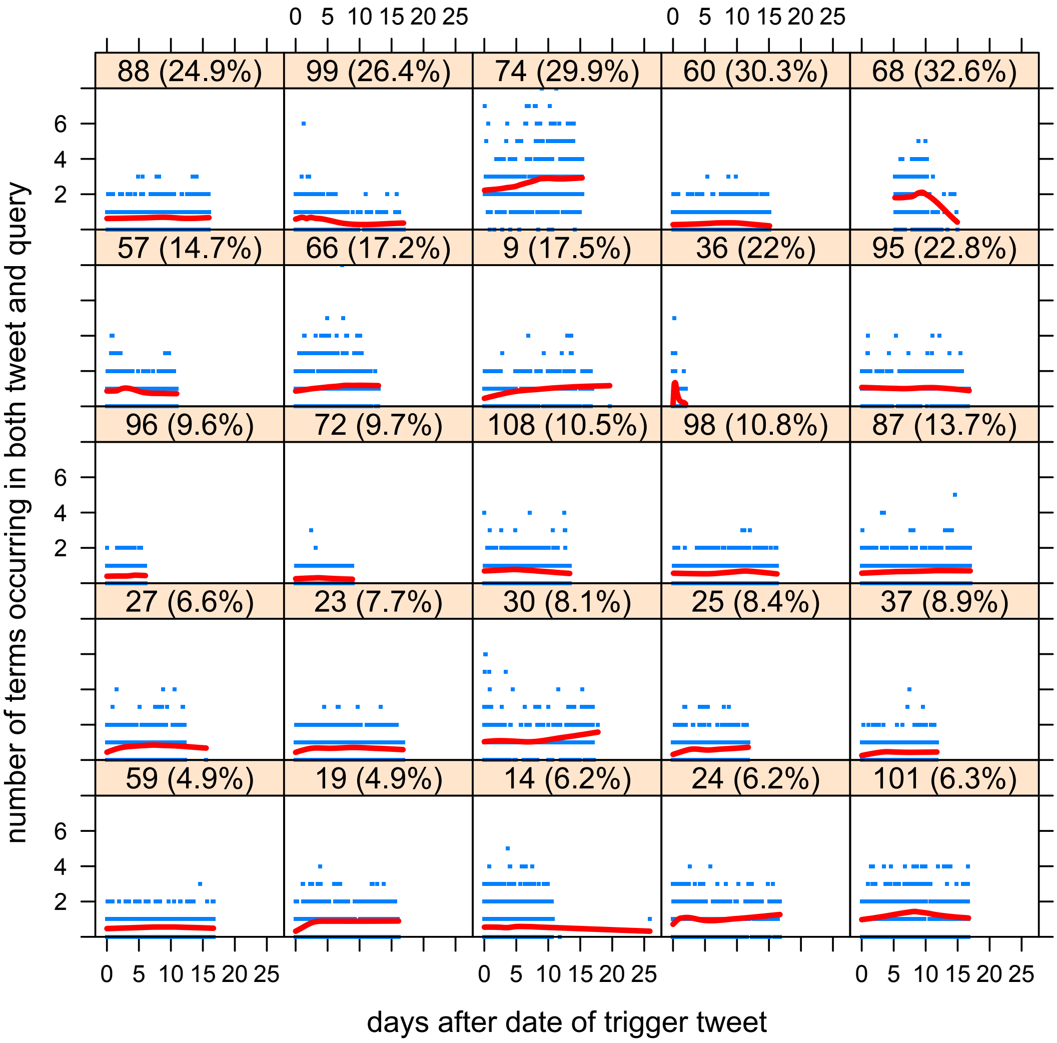


Figure 3.6: The number of similar terms in tweets and queries over days passed from the trigger tweet for different queries with high relevance

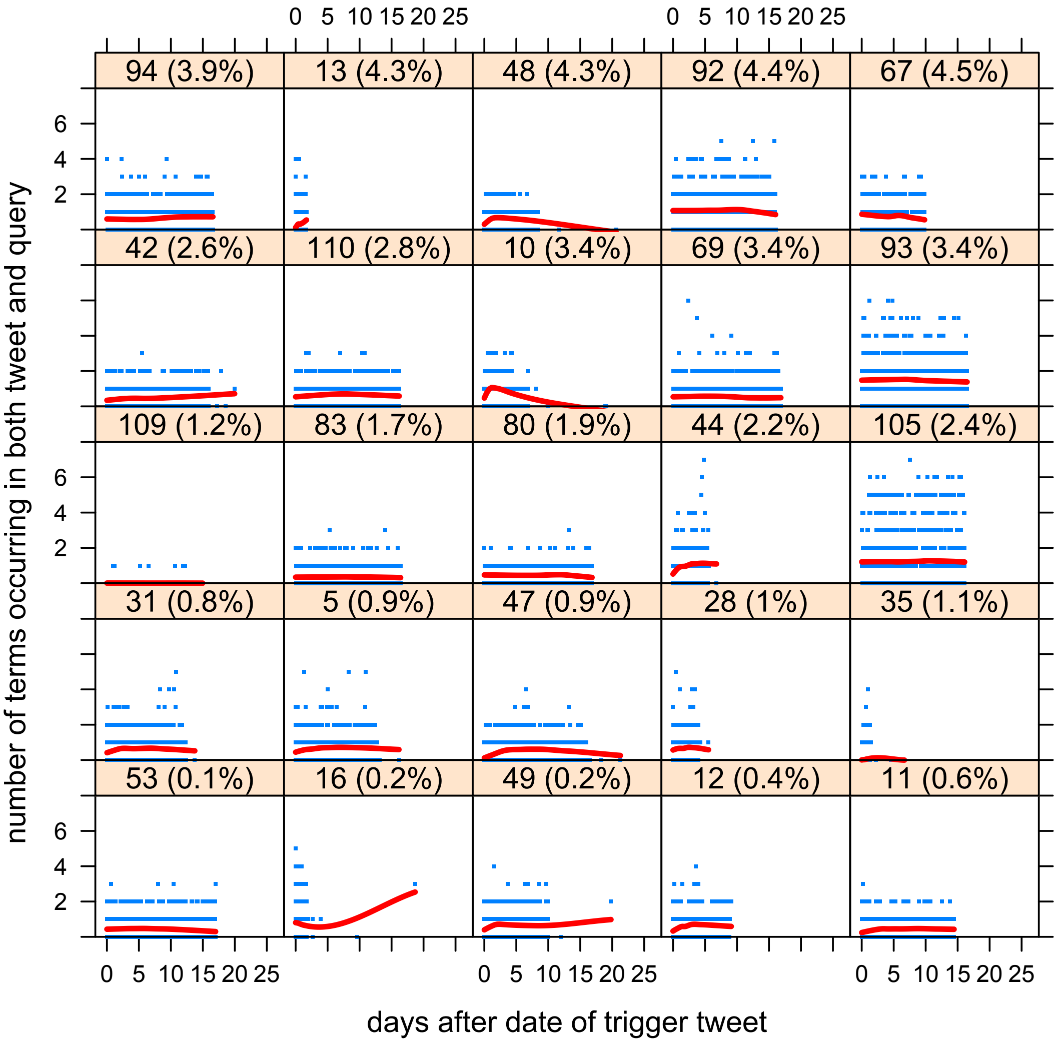


Figure 3.7: The number of similar terms in tweets and queries over days passed from the trigger tweet for different queries with low relevance

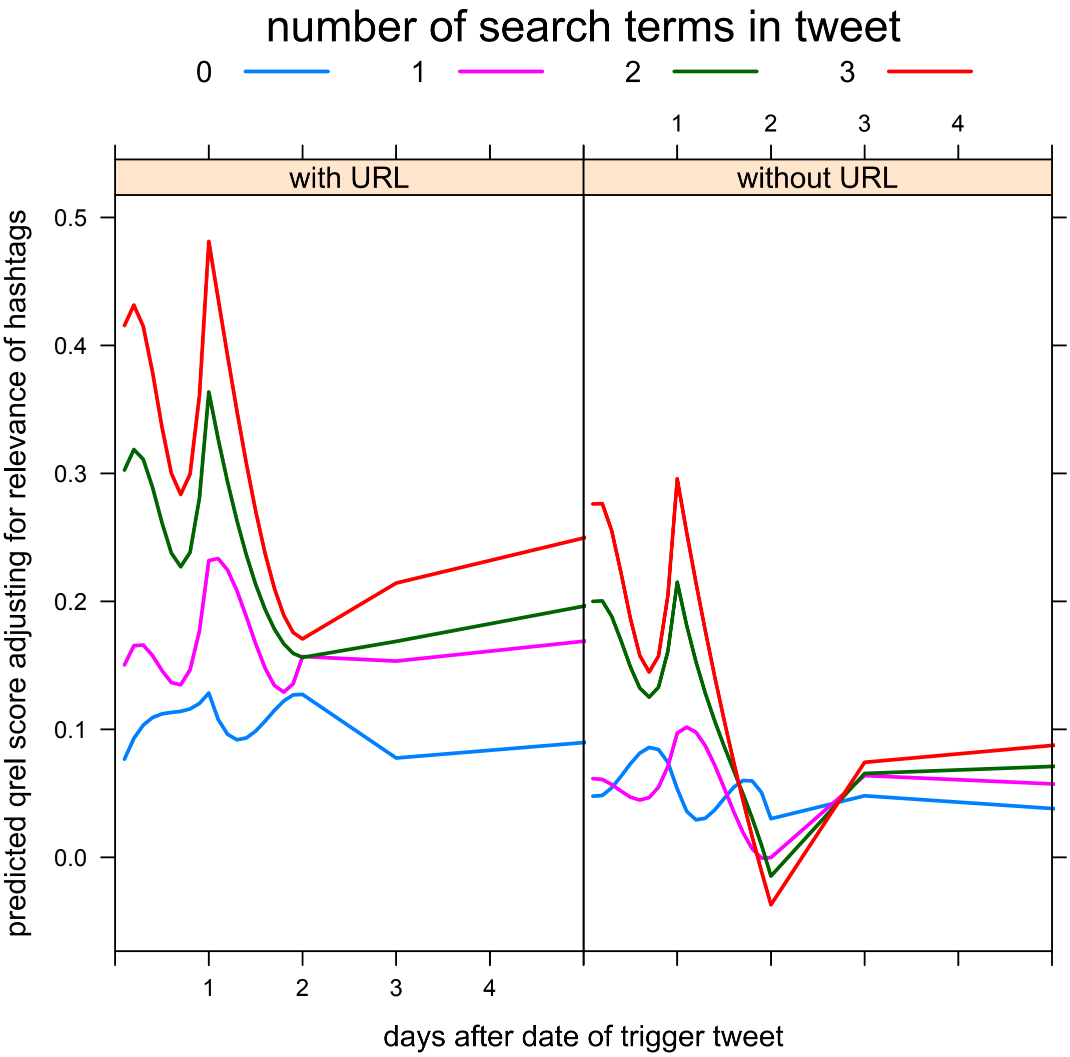


Figure 3.8: Tweet relevance over time for all tweets, both with URL and without URL

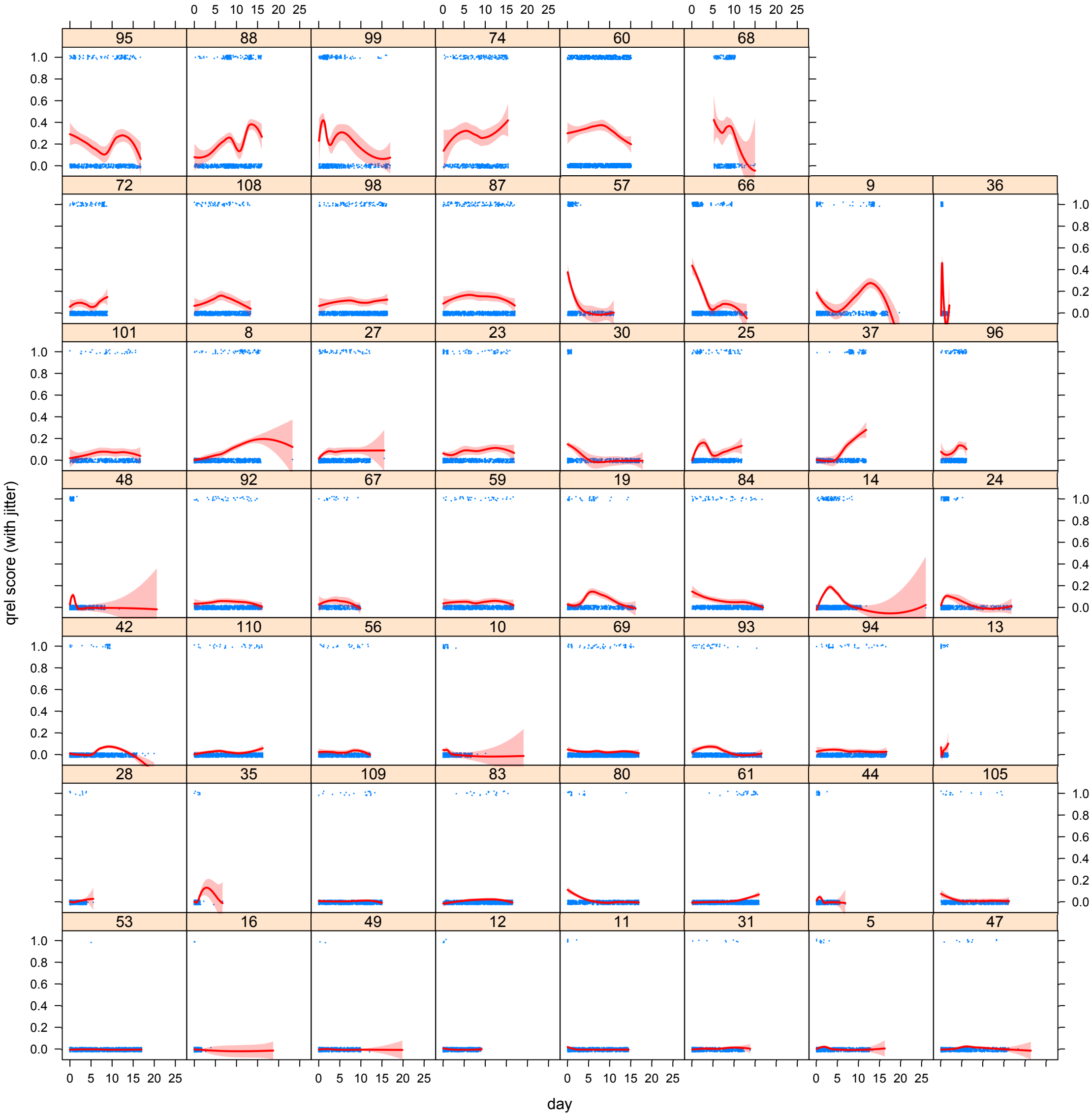


Figure 3.9: The effect of day in relevance between the queries.

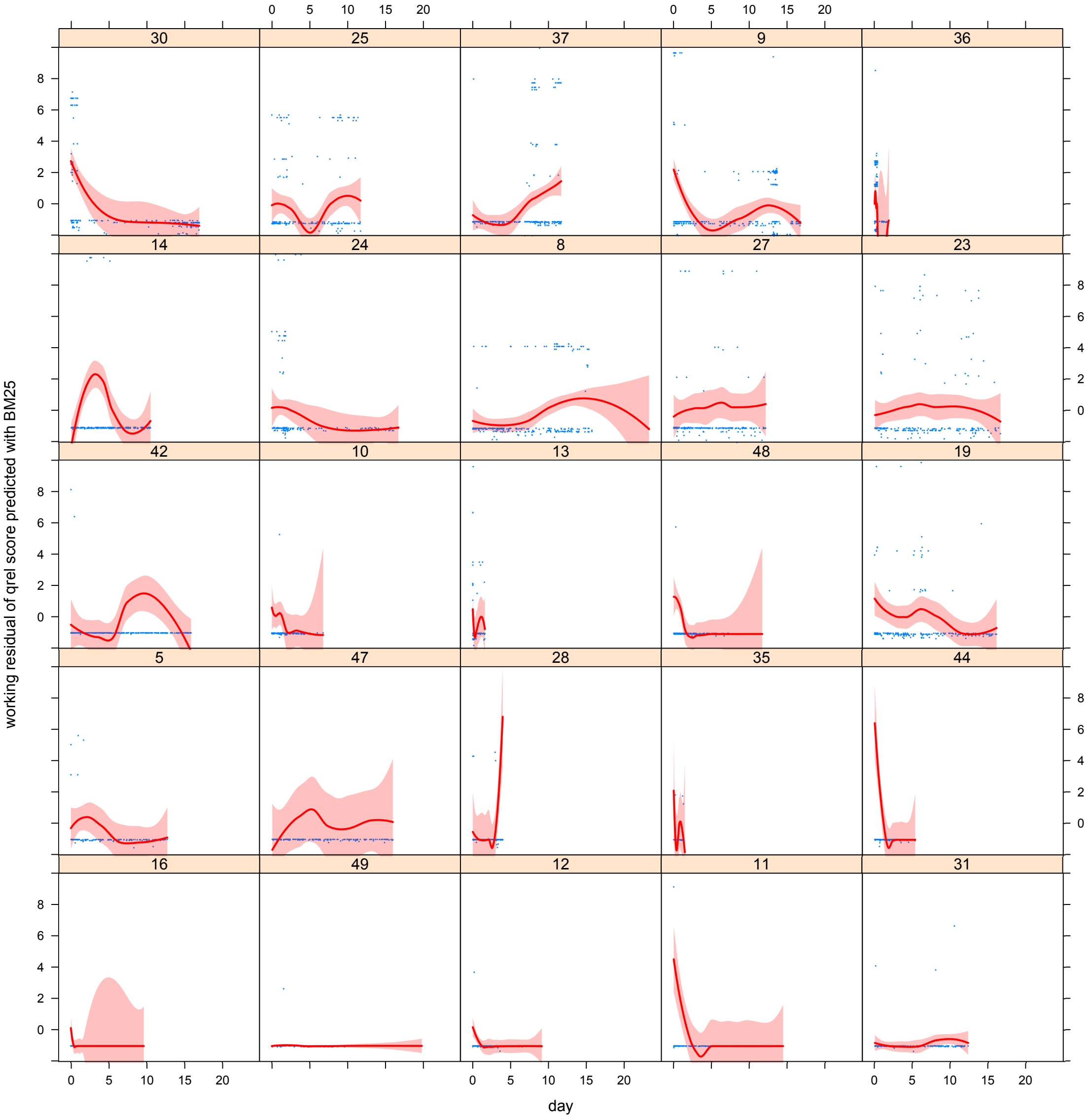


Figure 3.10: The working residual from BM25 versus day - partial residual plot

# 4 Proposed methods

With the rapid growth of the Internet, the amount of short text data on the Web is also growing. There are several models of short text, and microblog is one of the most popular ones. Twitter is a service providing microblog posting. The popularity of Twitter is growing among users, and correspondingly research in the field is attracting many scientists. Because of their nature, microblogs share many properties of social networks while lacking some properties of long text documents. One of the vastly noted properties of short text is a sparse feature space that makes it difficult to discover correlations among the features. Immediacy and being nonstandard are among the other most important features of short text [55]. Since microblogs are immediate, it leads to real-time generation of information and consequently a large quantity of the produced short text documents. On the other hand, the content is brief, misspelling is common and nonstandard language and structure is frequently used. The evaluation results elaborated in the microblog track of TREC suggest that an appropriate resolution to the real-time search task is yet to be found [34]. In this chapter, following the extended statistical analysis presented in previous chapters, models are introduced for microblog retrieval.

## 4.1 Logistic multivariate mixed model

To capture the importance of different features, with a focus on temporal features, a multivariate mixed model framework were chosen. The reason for this decision lies in the nature of the study and the nature variables that affect the weighting function. In other words, there are plenty of different factors and parameters that contribute, either positively or negatively, to the relevance of the microblog to the query. The response function in this study is relevance. I decided to simply classify the tweets into two groups: relevant or irrelevant. The criterion was human judgment included in the corpus[8]. Accordingly the response variable is binary for which logistic regression is the conventional approach. However, ordinary logistic regression is only appropriate for data in which the observations are mutually independent. It is not appropriate for data in which observations are naturally clustered in groups that share common unmeasured attributes. Such a clustered structure leads to a violation of the assumption of independence and the use of ordinary logistic regression is likely to produce biased parameter estimates and incorrect estimates of their standard errors as described in the case of linear models in [36]. In this data, the units of analysis are the tweets, which are clustered within queries. Multilevel

[8] As declared in Chapter 3 microblog track of TREC12 as well as TREC11 includes a set of tweets along with queries and a gold standard for relevance of tweets to queries.

methods that allow random variation between queries, as well as between tweets within queries address the shortcomings of ordinary logistic regression for this data. Generalized Linear Mixed Model methods [44] incorporate between query random effects as well as between tweet-within query random effects. Under assumptions discussed later they allow unbiased estimates of the coefficients relating relevance to various explanatory variables as well as correct estimates of their standard errors.

### 4.1.1 Logistic regression

Logistic regression is a special type of generalized linear regression that has been used in the current work. In this setting we have a binary output variable $rel$ , and we want to model the conditional probability $Pr(rel = 1|X = x)$ as a function of $x$; any unknown parameters in the function are to be estimated by maximum likelihood. In other words we want to model the probability of a tweet relevance as a function of different features introduced earlier in Chapter 3. Using a logistic regression is a way of solving the problem with the approach of linear regression. However a simple linear $p(x)$ cannot be useful since unlike a general linear function it is not unbounded and has to be between 0 and 1. The next idea would be to use log $p(x)$ as a linear function of $x$ but logarithms are unbounded in only one direction. This is not true for linear functions. The most straightforward alteration of $logp$ for eliminating this boundary is the logistic (or $logit$) transformation, $log\frac{p}{1-p}$ . We can make this a linear function of $x$ without fear of nonsensical results[12]. The model formally looks like 4.1.

$$logit(p(x)) = log\frac{p(x)}{1-p(x)} = \beta_0 + x_1.\beta_1 + ... + x_k.\beta_k \tag{4.1}$$

Solving this we will have the score as shown in equation 4.2

$$score = p(x; b, w) = \frac{e^{\beta_0+x_1.\beta_1+...+x_k.\beta_k}}{1+e^{\beta_0+x_1.\beta_1+...+x_k.\beta_k}} = \frac{1}{1+e^{-(\beta_0+x_1.\beta_1+...+x_k.\beta_k)}} \tag{4.2}$$

In our proposed model $x_i$s reperesent different features that affect the relevance of the tweet and $\beta_i$s are the coefficients. Since there are different features that can contribute to relatedness of a tweet to a particular query, using multi-level models seems to be an appropriate decision.

### 4.1.2 Generalized linear mixed models (GLMM)

Generalized Linear Mixed Models (GLMM) is a straightforward extension of the generalized linear model that adds random effects to the linear predictor. These random effects are assumed to have a normal distribution. GLMM adds random effects to the linear predictor, and expresses the expected value of the response conditional on the random effects. A generalized linear mixed model for the relevance of tweets consists of three components:

- a random variable $Y_{ij}$ reperesenting the relevence (0 or 1) of the $j^{\text{th}}$ tweet of the $i^{\text{th}}$ query., whose distribution is binomial with mean $\mu_{ij}$
- a link function, in this case a logit function that relates $\mu_{ij}$ to a linear predictor $\eta_{ij}$:

  $$\eta_{ij} = log(\frac{\mu_{ij}}{1-\mu_{ij}})$$

  or, equivalently,

  $$\mu_{ij} = \frac{e^{\eta_{ij}}}{1-e^{\eta_{ij}}}$$

- a linear model for the linear predictor

  $$\eta_i = \beta_0 + \beta_1.X_{1ij} + ... + \beta_k.X_{kij} + b_i.Z_{ij}$$

  where $X_{lij}$, $l = 1, ..., K$ denotes a predictor that is a characteristic of the $j$th tweet in the $i$th query ($X_{lij}$ may be a characteristic of the query only in which case it would be constant with respect to j), and a random cluster intercept $b_i$ that is assumed to have a normal distribution with mean 0 and variance $Z^2$, varying randomly from cluster to cluster [44].

## 4.2 Logistic multivariate mixed models for microblog re-ranking

Two models for re-ranking has been proposed based on 4.1. In order to formulate a model based on a series of analyses carried out on the learning set, some features were chosen to be included in the model. In addition to the features, some interactions were also considered. For some of the variables in the model, neither linear nor quadratic functions could describe their relation with the response variable, i.e. relevance. Consequently linear splines were used. Spline functions are piecewise polynomials with continuity constraints used in curve fitting. A set of knots divides the range of $X$ into intervals and within each interval a polynomial is used with possibly different degrees in different intervals. These polynomials are connected at the knots subject to continuity constraints. For example in the case of a spline with three knots $a <b <c$, a linear spline function with continuity at the knots is given by:

$$f(x) = \beta_0 + \beta_1 X + \beta_2 (X - a)_+ + \beta_3 (X - b)_+ + \beta_4 (X - c)_+, \tag{4.3}$$

Where;

$$\begin{aligned} (u)_+ &= u, \quad u > 0, \\ & \phantom{=} \ \ 0, \quad u < 0. \end{aligned} \tag{4.4}$$

Depending on patterns present in the data, a spline using polynomials of a higher degree may be appropriate [15]. In the rest of this section the re-ranking models are explained in more detail.

### 4.2.1 Model 1: Re-ranking with tweet-specific functions

The first model, which is shown in equation 4.5 takes advantage of features extracted from the tweet without any use of traditional information retrieval weighting functions.

The main reason for this choice was to investigate whether simply-extracted tweet-specific features can make up for shortcomings of tweet documents inherited by their short text nature. The most common approach is to expand the documents; however, adding extra features enlarges the the size of the data set and makes the task of retrieval less efficient for most of the existing approaches. The analysis undertaken provided a basis for choosing the features to be included in the model and logistic multivariate mixed model approach was used to estimate the coefficients. Table 4.1 displays the resulting estimates for coefficients in the model and their significance. According to the values in the table, *tweetLength*, *URLexist*, *BM25*, *mentionCount*, and *day* are significant. Neither a linear nor a quadratic function adequately captured the effect of *day* and *hashtagCount* therefore spline functions were used. The results show that three or more hashtags can affect the relevance positively until the number gets as high as eight.

After confirming the significance of the coefficients a Wald test is done to test the significance of the model. The results are shown in Table 4.3 and justify use of the model for this purpose.

I tested the need for inclusion of each of these terms by testing whether the coefficients relating to these terms could be dropped. Table 4.6 shows p-values to test the exclusion of the various terms in the model. In the Wald test for each term is tested whether all the coefficients that include that term could be removed from the model. The results suggest that all of the terms are significant and should remain in the model.

Table 4.1: Coefficientsand their significance for model 1

| **Variable** | **Estimate** | **Std. Error** | **z value** | $Pr(>\|z\|)$ |
|---|---|---|---|---|
| (Intercept) | -8.454 | 1.043 | -8.108 | 5.13e-16 *** |
| tweetLength | 0.003 | 0.0006 | 5.402 | 6.58e-08 *** |
| hashTagCount | -0.076 | 0.021 | -3.673 | 0.0002 *** |
| URLExist | 0.927 | 0.0494 | 18.739 | $<$ 2e-16 *** |
| mentionCount | -0.196 | 0.040 | -4.850 | 1.23e-06 *** |
| averageLength(0,4) | 0.765 | 0.263 | 2.910 | 0.0036 ** |
| averageLength(4,5) | -0.220 | 0.300 | -0.733 | 0.464 |
| averageLength(5+) | -0.540 | 0.088 | -6.115 | 9.68e-10 *** |
| day(0,1) | 0.840 | 0.125 | 6.704 | 2.02e-11 *** |
| day(1,6) | -0.896 | 0.137 | -6.543 | 6.02e-11 *** |
| day(6,13) | 0.094 | 0.032 | 2.900 | 0.004 ** |
| day(13+) | -0.244 | 0.064 | -3.835 | 0.0001 *** |
| numOccur | 1.122 | 0.072 | 15.551 | $<$ 2e-16 *** |
| numOccurHashTag | 1.726 | 0.290 | 5.959 | 2.55e-09 *** |
| day(0,1):numOccur | 0.774 | 0.095 | -8.155 | 3.50e-16 *** |
| day(1,6):numOccur | 0.771 | 0.104 | 7.421 | 1.16e-13 *** |
| day(6,13):numOccur | 0.006 | 0.023 | 0.268 | 0.789 |
| day(13+):numOccur | 0.002 | 0.047 | 0.046 | 0.963 |
| day(0,1):numOccurHashTag | -1.131 | 0.324 | -3.491 | 0.0005 *** |
| day(1,6):numOccurHashTag | 1.045 | 0.345 | 3.030 | 0.002 ** |
| day(6,13):numOccurHashTag | 0.172 | 0.085 | 2.026 | 0.043 * |
| day(13+):numOccurHashTag | -0.051 | 0.173 | -0.296 | 0.767 |

The resulting model 1 is presented in Equation 4.5.

Table 4.2: Overall Wald test for model 1 excluding the intercept

| DF | $\chi^2$ | p-value |
|---|---|---|
| 21 | 1366.71 | <0.00001 |

Table 4.3: Partial Wald tests for model 1

| Hypothesis | DF | $\chi^2$ | p-value |
|---|---|---|---|
| Overall interactions | 8 | 121.15 | <.00001 |
| Interactions between day and numOccur | 4 | 105.66 | <.00001 |
| Interactions between day and numOccurHashTag | 4 | 24.52 | 6e-05 |
| Overall test for day | 12 | 162.46 | <.00001 |
| Overall test for spline for day | 9 | 133.89 | <.00001 |
| Overall test for numOccur | 5 | 485.11 | <.00001 |
| Overall test for numOccurHashTag | 5 | 90.34 | <.00001 |
| Overall test for averageLength | 3 | 122.43 | <.00001 |
| URLExist | 1 | 351.13 | <.00001 |
| tweetLength | 1 | 29.17 | <.00001 |
| mentionCount | 1 | 23.52 | <.00001 |
| hashTagCount | 1 | 13.49 | 0.00024 |

Table 4.4: Coefficients and their significance for model 2

| Variable | Estimate | Std. Error | z value | $Pr(>\|z\|)$ | |
|---|---|---|---|---|---|
| (Intercept) | -4.842 | 0.357 | -13.553 | < 2e-16 | *** |
| tweetLength | -0.004 | 0.002 | -2.303 | 0.021 | * |
| URLExist | 0.977 | 0.125 | 7.802 | 6.10e-15 | *** |
| BM25 | 0.160 | 0.010 | 16.501 | < 2e-16 | *** |
| mentionCount | -0.356 | 0.134 | -2.648 | 0.008 | ** |
| day(0,1) | -1.261 | 0.217 | -5.819 | 5.94e-09 | *** |
| day(1,6) | 1.178 | 0.244 | 4.822 | 1.42e-06 | *** |
| day(6,13) | 0.226 | 0.074 | 3.051 | 0.002 | ** |
| day(13+) | -0.984 | 0.174 | -5.667 | 1.45e-08 | *** |
| hashTagCount(0,1) | -0.199 | 0.157 | -1.269 | 0.204 | |
| hashTagCount(1,3) | 0.077 | 0.285 | 0.272 | 0.786 | |
| hashTagCount(3,8) | 0.724 | 0.270 | 2.682 | 0.007 | ** |
| hashTagCount(8+) | -1.478 | 0.567 | -2.605 | 0.009 | ** |

Table 4.5: Overall Wald test for model 2 excluding the intercept

| DF | $\chi^2$ | p-value |
|---|---|---|
| 12 | 433.47 | <0.00001 |

Table 4.6: Partial Wald tests for model 2

| Hypothesis | DF | $\chi^2$ | p-value |
|---|---|---|---|
| spline for day | 4 | 97.75 | <0.00001 |
| spline for hashTagCount | 4 | 26.48 | 3e-05 |
| tweetLength | 1 | 5.28 | 0.02154 |
| URLExist | 1 | 60.84 | <0.00001 |
| BM25 | 1 | 272.15 | <0.00001 |
| mentionCount | 1 | 6.99 | 0.00817 |

$$\begin{aligned}
logit(p) = &- 8.454 + 0.003 \times \text{tweetLength} - 0.076 \times \text{hashTagCount} \\
&+ 0.927 \times \text{URLExist} - 0.196 \times \text{mentionCount} + 1.726 \times \text{numOccurHashTag} \\
&+ 0.765 \times (\text{averageLength} - 0)_+ - 0.220 \times (\text{averageLength} - 4)_+ \\
&- 0.540 \times (\text{averageLength} - 5)_+ + 0.840 \times (\text{day} - 0)_+ - 0.896 \times (\text{day} - 1)_+ \\
&+ 0.094 \times (\text{day} - 6)_+ - 0.244 \times (\text{day} - 13)_+ + 1.122 \times \text{numOccur} \\
&- 0.774 \times \text{numOccur} \times (\text{day} - 0)_+ + 0.771 \times \text{numOccur} \times (\text{day} - 1)_+ \\
&+ 0.0062 \times \text{numOccur} \times (\text{day} - 6)_+ + 0.002 \times \text{numOccur} \times (\text{day} - 13)_+ \\
&- 1.131 \times \text{numOccurHashTag} \times (\text{day} - 0)_+ \\
&+ 0.944 \times \text{numOccurHashTag} \times (\text{day} - 1)_+ \\
&+ 0.172 \times \text{numOccurHashTag} \times (\text{day} - 6)_+ \\
&- 0.051 \text{numOccurHashTag} \times (\text{day} - 13)_+
\end{aligned} \tag{4.5}$$

Where;

$$(x)_+ = \begin{cases} 0, & \text{if } x < 0, \\ \\ 1, & \text{otherwise.} \end{cases}$$

thus

$$(x - c)_+ = max(0, x - c) \tag{4.6}$$

### 4.2.2 Model 2: Re-ranking with a BM25 microblog-customized approach

As explained in the last section, the first model uses tweet-specific features only. The second model on the other hand tries to improve the existing weighting functions for conventional retrieval purposes. In order to develop this model, a retrieval task was conducted first and the results for a baseline model were generated. The outcome weighting scores are represented as the variable BM25. Similar to the first model, we first investigated the significance of the coefficients and then that of the model, in Table 4.4 and 4.5 respectively.

$$
\begin{aligned}
logit(p) = & -4.842 - 0.004 \times \text{tweetLength} + 0.977 \times \text{URLExist} \\
& + 0.160 \times \text{BM25} - 0.356 \times \text{mentionCount} - 1.261 \times (\text{day} - 0)_+ \\
& + 1.178 \times (\text{day} - 1)_+ + 0.225975 \times (\text{day} - 6)_+ - 0.984 \times (\text{day} - 13)_+ \\
& - 0.199 \times (\text{hashTagCount} - 0)_+ + 0.077 \times (\text{hashTagCount} - 1)_+ \\
& + 0.724 \times (\text{hashTagCount} - 3)_+ - 1.478 \times (\text{hashTagCount} - 8)_+
\end{aligned}
\tag{4.7}
$$

Where;

$$
(x)_+ = \begin{cases} 0, & \text{if } x < 0, \\ 1, & \text{otherwise.} \end{cases}
$$

thus

$$
(x - c)_+ = max(0, x - c) \tag{4.8}
$$

## 4.3 A hybrid retrieval model for microblog retrieval

For filtering we mainly evaluate a recently proposed hybrid retrieval model [52] which has shown to be very effective on a large number of TREC data sets for ad hoc information retrieval.

In particular, this hybrid model extends Rocchio's feedback method by incorporating three kinds of IR techniques, which are proximity, feedback document quality estimation and query performance prediction techniques, under the pseudo relevance feedback (PRF) framework to boost the overall performance. In our experiments, we test different settings of this hybrid model on the microblog data set. In the rest of this section, we briefly describe this hybrid model. Rocchio's algorithm [38] is a classic framework for implementing (pseudo) relevance feedback via improving the query representation. Although Rocchio's model has been introduced many years ago, it is still effective in obtaining relevant documents. According to [53], "BM25 term weighting coupled with Rocchio feedback remains a strong baseline which is at least as competitive as any language modeling approach for many tasks". However, the traditional form of Rocchio's model can still be reformed to be better. First, the query term proximity information which has proven to be useful is not considered. Second, Rocchio's algorithm views terms from different feedback documents equally. Intuitively, a candidate expansion term in a document with better quality is more likely to be relevant to the query topic. Third, the interpolation parameter $\alpha$ is always fixed across a group of queries.

In order to address these problems, Ye *et al.* [52] extend Rocchio's algorithm by refining the query representation as follows:

$$Q_1 = \alpha * (\beta * Q_0 + (1 - \beta) * Q_p) + (1 - \alpha) * \sum_{r \in R} \frac{r * q(d_r)}{|R|} \tag{4.9}$$

where $\beta$ controls how much we rely on the query term proximity information [46], $\alpha$ controls how much we rely on the original query, $Q_p$ is an n-gram of original query terms and $q(d_r)$ is the quality score of document $d$. As we can see from Equation 4.9, this hybrid model is very flexible and can evaluate different techniques. In our experiments, we adopt the co-occurrence interpretation of term proximity to compute $Q_p$, where the proximity among query terms is represented by the n-gram frequencies and BM25 is used as the weighting model [16]. Full dependencies of query terms are taken into account. For the document quality factor $q(d_r)$, we simply use the normalized scores from the first-pass retrieval for approximation as described in [51]. For the term weighting formula in the query expansion component, we simply use the Lemur[9] TFIDF formula, which was shown to be surprisingly effective on a number of standard TREC collections in our preliminary experiments.

As we can see from Equation 4.9, testing different combinations of the component techniques is a straightforward process. In the following, we summarize the component models and the corresponding setting of parameters for the four different runs of the experiments.

Table 4.7: The settings of our submissions

| Run | Basic model | Proximity Model | QE Model |
|---|---|---|---|
| YORK1 | DFRee | NO | KL weighting Model ($doc$= 20, $term$=30, $\beta$=1.4) |
| YORK2 | BM25 ($b$= 0.3) | NO | KL weighting Model ($doc$= $20term$=$30\alpha$=1.4) |
| york12mb3 | DFRee | NO | KL weighting Model ($doc$= $20term$=$30\alpha$=1) |
| york12mb4 | DFRee | Yes ($\beta$=0.1,$wSize$=8) | KL weighting Model ($doc$= $20term$=$30\alpha$=1) |

### 4.3.1 Parameters

We empirically set parameters as follows; $\alpha$ to 0.6, $b$ in BM25 to 0.3 and $\beta$ to be 0.2. We did not use the proximity model in run 3, while all the components were used in run 4 with the parameters setting described above.

### 4.3.2 A modified BM25

In order to take the specific features of Twitter social network into account we changed BM25 weighting model for the YORK2 and york12mb3 runs. The new model linearly combines four different scores as follows:

$$Score(T, D) = w_1 * Score1 + w_2 * Score2 + w_3 * QM_1 + w_4 * QM_2 \tag{4.10}$$

[9] Lemur Toolkit is an open-source software framework for building language modeling and information retrieval software.

where $w_i$ is a real number and

$$w_2 > w_1 > w_3 > w_4 \tag{4.11}$$

The first term of Equation 4.10 uses the traditional BM25 score. BM25 calculates the score as shown below:

$$Score1 = \sum_{(q_i \in Q)} \frac{f(q_i, d)}{k_1 * \left((1-b) + b * \frac{|D|}{|avgdl|}\right) + f(q_i, d)} * idf(q_i) \tag{4.12}$$

The other three terms consider hashtags and links in the tweets. Twitter help center[10] recommends Twitter users not to use more than two hashtags in each tweet. We investigated hashtags in two cases: first, when the hashtag term exists in the topic query and, second, for the cases where it does not match query terms. The former is weighted using a similar formulation as BM25 as follows if $\exists h_i \in Q$:

$$Score2 = \sum_{(q_i \in Q)} \frac{f(h_i, d)}{k_1 * \left((1-b) + b * \frac{|D|}{|avgdl|}\right) + f(h_i, d)} * idf(h_i) \tag{4.13}$$

For the times that hashtag terms do not occur in the topic we only consider the fre-

[10] https://support.twitter.com/articles/49309-what-are-hashtags-symbols

quency. This was implemented as shown in Equations 4.14 and 4.15.

$$QM_1 = logP(h|D) \tag{4.14}$$

$$P(h|D) = \begin{cases} 1 & n_h > 0 \\ 0 & n_h = 0 \end{cases} \tag{4.15}$$

Since we did not use any external evidence in our experiments, we did not take the content of URLs into consideration. Equations 4.16 and 4.17 dedicate a positive score to tweets with URLs.

$$QM_2 = logP(l|D) \tag{4.16}$$

$$P(l|D) = \begin{cases} 1 & n_l > 0 \\ 0 & n_l = 0 \end{cases} \tag{4.17}$$

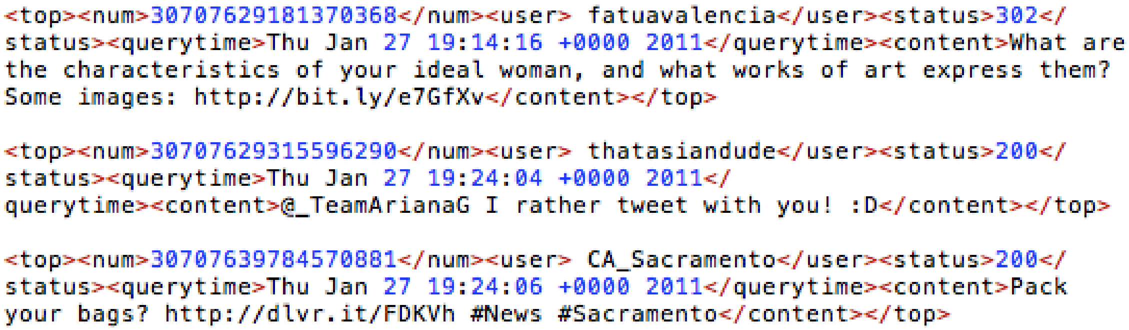
All the coefficients in 4.10, i.e. $w_i$s, were tuned using microblog track 2011 dataset as training data.

# 5 Experimental Setting

## 5.1 Data set

Our experiments make use of the Text Retrieval Conference (TREC) 2011 Microblog Dataset, also known as the Tweets2011 Corpus or simply just Tweets11, as an initial source for tweet messages. The corpus consists of approximately 16 million microblog posts, or tweets, made available by the online social networking and microblogging service Twitter. Tweets are 140-character status updates that can refer about any multitude of different topics. These tweets were collected over a period of two weeks in 2011 from January 24th to February 8th inclusive. This is a diverse period covering both the Egyptian revolution and the 45th Super Bowl among other topics and events. The corpus is intended to be a reusable, representative sample of the twittersphere[11] and different types of tweets are included such as replies and retweets. This dataset was selected for use with our experiment as it exhibited the distinct characteristics traditionally associated with short text including being immediate, noisy, of a minimal number of characters, and

[11] http://trec.nist.gov/data/tweets/



```
<top><num>30707629181370368</num><user> fatuavalencia</user><status>302</
status><querytime>Thu Jan 27 19:14:16 +0000 2011</querytime><content>What are
the characteristics of your ideal woman, and what works of art express them?
Some images: http://bit.ly/e7GfXv</content></top>

<top><num>30707629315596290</num><user> thatasiandude</user><status>200</
status><querytime>Thu Jan 27 19:24:04 +0000 2011</
querytime><content>@_TeamArianaG I rather tweet with you! :D</content></top>

<top><num>30707639784570881</num><user> CA_Sacramento</user><status>200</
status><querytime>Thu Jan 27 19:24:06 +0000 2011</querytime><content>Pack
your bags? http://dlvr.it/FDKVh #News #Sacramento</content></top>
```

Figure 5.1: A fraction of TREC microblog11 data set

containing few features. Each day of the corpus is split into files called blocks, each of which contains about 10,000 tweets compressed using gzip [32]. These status blocks were fetched from twitter.com using an HTML crawler. Using twitter-corpus-tools, the corpus was read and the output of each block was copied to an individual .html file. The tweets within the HTML files have one of five status codes associated with them, namely 200, 302, 403, 404, and null, which designate ok, found, forbidden, not found and nothing respectively. The number and status of the tweets varies based on the crawling time, the network environment, and other possible factors [25]. A fraction of the tweets data set is shown in Figure 5.1.

In our case, the corpus was fetched via the corpus downloader included 5,850,415 tweets that were shown to be forbidden, not found, or indicated to be nothing  as such, these were ultimately not included. These HTML files were parsed during a previous experiment and combined into one large file. The enclosed tweets were also encapsulated with XML tags in order to distinguish between individual tweets and the internal elements that make up their structure. Given that the nature of the TREC Microblog Tracks task is that of information retrieval, the Tracks organizers created fifty test topics in 2011 and accumulated relevance judgments in a fashion similar to the standard TREC pooling method [11]. In 2012, 60 more topics were added in a similar manner. TREC microblog 2011, or simply tweets11, data set is used in the experiments presented in the current work. This data set is accompanied with two sets of query topic lists. TREC experts carefully design these query topic lists in the years 2011 and 2012. In the following chapter, the corpus will be discussed thoroughly[12].

### 5.1.1 Part1: Tweets

The tweets11 data set consists of 16 million tweets of which about 6 million are posted in languages other than English. In the current experiments two subsets are driven from those tweets that are written in English: A training data set and a test data set. Since the human judgments provided by expert assessors, or qrels, play a significant role in the experiments, tweets that are associated with qrel scores are included in both data sets. To distinguish between training and test sets, half of the query topics were picked randomly to have the tweets related with them included in the training set. This process is done using R and the code can be found in Appendix A.

[12] The java code for data extraction and configuration is included in Appendix B.

### 5.1.2 Part2: Queries

As mentioned earlier there are two sets of queries each for 2011 and 2012. The first set includes 50 topic queries and the second consists of 60 topic queries. Table 5.1 and 5.2 list the query topics of 2011 data set. In addition to the content of each query, the query time, the number of terms, and the average term length are extracted and listed in these tables. The query time is a measure used for determining the time variable used throughout the experiments. The time variable of the queries provided by TREC is in the format ”day month date hr:min:sec year” as seen in the tables below. In order to use the time variable in the calculations of the weighting scores, the time is converted into a real number. The java code for this is included in Appendix B.

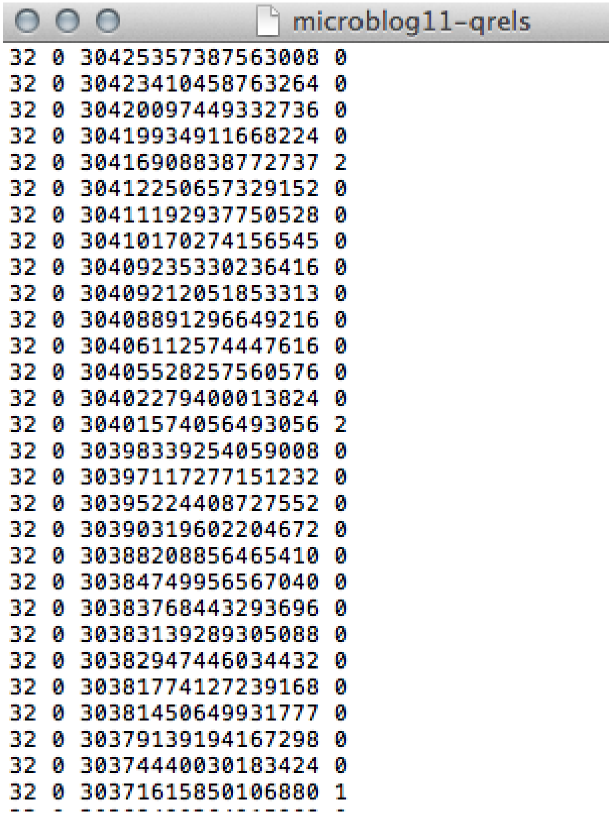
Tables 5.3 and 5.4 contain query topics for 2012 and the associated variables.

### 5.1.3 Part3: Gold Standard

The relevance judgment provided by TREC is in the form of a matrix, which is called a qrel file. The matrix provides tweet ids of the top 1000 retrieved tweets and their corresponding relevance scores. Human assessors have given a relevance score to each tweet based on its relevance to the query. If a tweet is irrelevant it gets a 0 score. 1 and 2 relevance scores show that the tweet is relevant or highly relevant respectively. There is a chance that the tweet is spam, which has been indicated with -2 score. The qrel

Table 5.1: Queries in the data set for 2011- part 1

| query id | Content | Query Time | Number of Terms | Average term length |
|---|---|---|---|---|
| 1 | BBC World Service staff cuts | Tue Feb 08 12:30:27 2011 | 5 | 4.8 |
| 2 | 2022 FIFA soccer | Tue Feb 08 18:51:44 2011 | 3 | 4.67 |
| 3 | Haiti Aristide return | Tue Feb 08 21:32:13 2011 | 3 | 6.33 |
| 4 | Mexico drug war | Wed Feb 02 17:22:14 2011 | 3 | 4.33 |
| 5 | NIST computer security | Fri Feb 04 17:44:09 2011 | 3 | 6.67 |
| 6 | NSA | Tue Feb 08 16:00:59 2011 | 1 | 3 |
| 7 | Pakistan diplomat arrest murder | Tue Feb 08 22:56:33 2011 | 4 | 6.75 |
| 8 | phone hacking British politicians | Mon Feb 07 17:42:59 2011 | 4 | 7.5 |
| 9 | Toyota Recall | Tue Feb 08 21:41:26 2011 | 2 | 12 |
| 10 | Egyptian protesters attack museum | Sat Jan 29 20:06:35 2011 | 4 | 7.5 |
| 11 | Kubica crash | Sun Feb 06 10:38:43 2011 | 2 | 5.5 |
| 12 | Assange Nobel peace nomination | Mon Jan 31 21:02:33 2011 | 4 | 6.5 |
| 13 | Oprah Winfrey half-sister | Mon Jan 24 15:43:41 2011 | 3 | 7.67 |
| 14 | ”release of ””The Rite””” | Wed Feb 02 12:31:02 2011 | 4 | 4.5 |
| 15 | Thorpe return in 2012 Olympics | Sun Jan 30 12:20:25 2011 | 5 | 5.2 |
| 16 | ”release of ””Known and Unknown””” | Mon Jan 24 17:03:52 2011 | 5 | 5.2 |
| 17 | White Stripes breakup | Wed Feb 02 19:13:40 2011 | 3 | 6 |
| 18 | William and Kate fax save-the-date | Wed Jan 26 08:59:32 2011 | 5 | 6 |
| 19 | Cuomo budget cuts | Mon Feb 07 23:25:02 2011 | 3 | 5 |
| 20 | Taco Bell filling lawsuit | Sun Feb 06 07:09:20 2011 | 4 | 5.5 |
| 21 | Emanuel residency court rulings | Sat Jan 29 03:03:30 2011 | 4 | 7 |
| 22 | healthcare law unconstitutional | Tue Feb 01 22:17:34 2011 | 3 | 9.67 |
| 23 | Amtrak train service | Tue Feb 08 20:04:25 2011 | 3 | 6 |
| 24 | ”Super Bowl, seats” | Tue Feb 08 17:11:04 2011 | 3 | 5 |
| 25 | TSA airport screening | Thu Feb 03 19:52:09 2011 | 3 | 6.67 |



```
32 0 30425357387563008 0
32 0 30423410458763264 0
32 0 30420097449332736 0
32 0 30419934911668224 0
32 0 30416908838772737 2
32 0 30412250657329152 0
32 0 30411192937750528 0
32 0 30410170274156545 0
32 0 30409235330236416 0
32 0 30409212051853313 0
32 0 30408891296649216 0
32 0 30406112574447616 0
32 0 30405528257560576 0
32 0 30402279400013824 0
32 0 30401574056493056 2
32 0 30398339254059008 0
32 0 30397117277151232 0
32 0 30395224408727552 0
32 0 30390319602204672 0
32 0 30388208856465410 0
32 0 30384749956567040 0
32 0 30383768443293696 0
32 0 30383139289305088 0
32 0 30382947446034432 0
32 0 30381774127239168 0
32 0 30381450649931777 0
32 0 30379139194167298 0
32 0 30374440030183424 0
32 0 30371615850106880 1
```

Figure 5.2: A snapshot of the qrel file indicating relevancy of tweets for query 32 (State of the Union and jobs )

file provides us with an evaluated subset of the twitter data set. This subset is divided into two data sets randomly providing a training and a testing data set. The analyses use only the training set, while the evaluations are executed on the test set. The evaluation is provided by experts and is included in the data set.

Table 5.2: Queries in the data set for 2011- part 2

| query id | Content | Query Time | Number of Terms | Average term length |
|---|---|---|---|---|
| 26 | US unemployment | Fri Feb 04 14:10:51 2011 | 2 | 7 |
| 27 | reduce energy consumption | Fri Feb 04 04:19:58 2011 | 3 | 7.67 |
| 28 | Detroit Auto Show | Wed Jan 26 22:46:12 2011 | 3 | 5 |
| 29 | global warming and weather | Tue Feb 08 01:05:57 2011 | 4 | 5.75 |
| 30 | Keith Olbermann new job | Tue Feb 08 22:51:01 2011 | 4 | 5 |
| 31 | Special Olympics athletes | Fri Feb 04 08:44:02 2011 | 3 | 7.67 |
| 32 | State of the Union and jobs | Fri Feb 04 02:08:22 2011 | 6 | |
| 33 | Dog Whisperer Cesar Millan's techniques | Thu Jan 27 19:27:54 2011 | 5 | 7 |
| 34 | MSNBC Rachel Maddow | Fri Feb 04 22:42:20 2011 | 3 | 5.67 |
| 35 | Sargent Shriver tributes | Mon Jan 24 07:18:17 2011 | 3 | 7.33 |
| 36 | Moscow airport bombing | Mon Jan 24 23:00:35 2011 | 3 | 5.67 |
| 37 | Giffords' recovery | Thu Feb 03 18:05:03 2011 | 2 | 8.5 |
| 38 | protests in Jordan | Tue Feb 01 12:46:40 2011 | 3 | 5.33 |
| 39 | Egyptian curfew | Fri Jan 28 18:14:09 2011 | 2 | 7 |
| 40 | Beck attacks Piven | Mon Jan 31 20:33:37 2011 | 3 | 5.33 |
| 41 | Obama birth certificate | Mon Jan 31 17:55:54 2011 | 3 | 7 |
| 42 | Holland Iran envoy recall | Mon Feb 07 20:47:13 2011 | 4 | 5.5 |
| 43 | Kucinich olive pit lawsuit | Sat Jan 29 08:06:05 2011 | 4 | 5.75 |
| 44 | White House spokesman replaced | Fri Jan 28 13:35:45 2011 | 4 | |
| 45 | political campaigns and social media | Tue Feb 01 12:52:29 2011 | 5 | 6.4 |
| 46 | Bottega Veneta | Tue Feb 08 22:34:59 2011 | 2 | 6.5 |
| 47 | organic farming requirements | Tue Feb 08 00:12:47 2011 | 3 | 8.67 |
| 48 | Egyptian evacuation | Mon Jan 31 09:36:57 2011 | 2 | 9 |
| 49 | carbon monoxide law | Tue Feb 01 22:44:23 2011 | 3 | 5.67 |
| 50 | ”war prisoners, Hatch Act” | Tue Jan 25 02:13:11 2011 | 4 | 5.5 |

Table 5.3: Queries in the data set for 2012- part 1

| query id | Content | Query Time | Number of Terms | Average term length |
|---|---|---|---|---|
| 51 | British Government cuts | Tue Feb 08 23:56:46 2011 | 3 | 7 |
| 52 | Bedbug epidemic | Thu Feb 03 16:24:58 2011 | 2 | 7 |
| 53 | river boat cruises | Tue Feb 08 23:51:47 2011 | 3 | 5.33 |
| 54 | The Daily | Tue Feb 08 04:49:44 2011 | 2 | 4 |
| 55 | berries and weight loss | Tue Feb 08 01:14:11 2011 | 4 | 5 |
| 56 | Hugo Chavez | Fri Feb 04 04:05:08 2011 | 2 | 5 |
| 57 | Chicago blizzard | Wed Feb 02 21:53:06 2011 | 2 | 7.5 |
| 58 | FDA approval of drugs | Tue Feb 08 18:31:56 2011 | 4 | 4.5 |
| 59 | Glen Beck | Tue Feb 08 16:31:32 2011 | 2 | 4 |
| 60 | fishing guidebooks | Mon Feb 07 02:22:19 2011 | 2 | 8.5 |
| 61 | Hu Jintao visit to the United States | Mon Feb 07 19:29:22 2011 | 7 | 4.28 |
| 62 | Starbucks Trenta cup | Tue Feb 08 23:02:21 2011 | 3 | 6 |
| 63 | Bieber and Stewart trading places | Sat Feb 05 20:34:21 2011 | 5 | 5.8 |
| 64 | red light cameras | Tue Feb 08 15:50:43 2011 | 3 | 5 |
| 65 | Michelle Obama's obesity campaign | Mon Feb 07 02:30:59 2011 | 4 | 7.5 |
| 66 | Journalists' treatment in Egypt | Sat Feb 05 00:32:03 2011 | 4 | 7 |
| 67 | Boston Celtics championship | Mon Feb 07 00:52:58 2011 | 3 | 8.33 |
| 68 | Charlie Sheen rehab | Tue Feb 01 20:14:53 2011 | 3 | 5.67 |
| 69 | high taxes | Tue Feb 08 20:34:23 2011 | 2 | 4.5 |
| 70 | farmers markets opinions | Mon Feb 07 21:44:55 2011 | 3 | 7.33 |
| 71 | Australian Open Djokovic vs. Murray | Sun Jan 30 12:36:20 2011 | 5 | 6.2 |
| 72 | Kardashians opinions | Mon Jan 31 21:22:35 2011 | 2 | 9.5 |
| 73 | Iran nuclear program | Fri Feb 04 12:41:50 2011 | 3 | 6 |
| 74 | credit card debt | Mon Feb 07 07:09:46 2011 | 3 | 4.67 |
| 75 | Aguilera super bowl fail | Tue Feb 08 21:56:22 2011 | 4 | 5.25 |
| 76 | Celebrity DUI violations | Tue Feb 08 10:34:12 2011 | 3 | 7.33 |
| 77 | NCIS | Mon Feb 07 11:53:05 2011 | 1 | 4 |
| 78 | McDonalds food | Tue Feb 08 20:06:32 2011 | 2 | 6.5 |
| 79 | Saleh Yemen overthrow | Fri Feb 04 20:03:25 2011 | 3 | 6.33 |
| 80 | Chipotle raid | Tue Feb 08 20:07:01 2011 | 2 | 6 |

Table 5.4: Queries in the data set for 2012- part 2

| query id | Content | Query Time | Number of Terms | Average term length |
|---|---|---|---|---|
| 81 | smartphone success | Tue Feb 08 05:29:10 2011 | 2 | 8.5 |
| 82 | illegal immigrant laws | Thu Feb 03 15:49:51 2011 | 3 | 6.67 |
| 83 | Stuxnet Worm effects | Tue Feb 08 09:57:04 2011 | 3 | 6 |
| 84 | Athlete concussions | Tue Feb 08 19:39:44 2011 | 2 | 9 |
| 85 | Best Buy improve sales | Tue Feb 08 00:05:56 2011 | 4 | 4.75 |
| 86 | Joanna Yeates murder | Mon Jan 31 14:01:16 2011 | 3 | 6 |
| 87 | chicken recipes | Tue Feb 08 23:16:17 2011 | 2 | 7 |
| 88 | Kings’ Speech awards | Tue Feb 08 00:48:24 2011 | 3 | 6 |
| 89 | Supreme Court cases | Thu Feb 03 02:25:14 2011 | 3 | 5.67 |
| 90 | anti-bullying | Tue Feb 08 10:27:39 2011 | 1 | 13 |
| 91 | Michelle Obama fashion | Sat Jan 29 23:21:02 2011 | 3 | 6.67 |
| 92 | stock market tutorial | Tue Feb 08 01:09:14 2011 | 3 | 6.33 |
| 93 | fashion week in NYC | Tue Feb 08 11:26:12 2011 | 4 | 4 |
| 94 | horse race betting | Tue Feb 08 13:46:36 2011 | 3 | 5.33 |
| 95 | Facebook privacy | Tue Feb 08 21:49:25 2011 | 2 | 7.5 |
| 96 | Sundance attendees | Sat Jan 29 00:29:48 2011 | 2 | 8.5 |
| 97 | college student aid | Tue Feb 08 22:18:57 2011 | 3 | 5.67 |
| 98 | Australian floods | Tue Feb 08 06:08:54 2011 | 2 | 8 |
| 99 | Superbowl commercials | Tue Feb 08 22:41:35 2011 | 2 | 10 |
| 100 | Republican National Committee | Mon Feb 07 22:18:19 2011 | 3 | 9 |
| 101 | ”Natalie Portman” in ”Black Swan” | Tue Feb 08 16:49:16 2011 | 5 | 6 |
| 102 | school lunches | Tue Feb 08 19:06:48 2011 | 2 | 6.5 |
| 103 | Tea Party caucus | Mon Feb 07 23:58:56 2011 | 3 | 4.67 |
| 104 | texting and driving | Mon Feb 07 15:40:10 2011 | 3 | 5.67 |
| 105 | The Avengers | Tue Feb 08 02:39:28 2011 | 2 | 5.5 |
| 106 | Steve Jobs’ health | Tue Feb 08 10:05:10 2011 | 3 | 5.33 |
| 107 | Somalian piracy | Tue Feb 08 11:48:46 2011 | 2 | 7 |
| 108 | identity theft protection | Sat Feb 05 09:46:43 2011 | 3 | 7.67 |
| 109 | Gasland | Sun Feb 06 23:16:42 2011 | 1 | 7 |
| 110 | economic trade sanctions | Tue Feb 08 04:58:53 2011 | 3 | 7.33 |

# 6 Experiments

Following the methodologies explained in chapter 4, and using the models developed some experiments were conducted to test performances of these models. The first part of the present chapter deals with evaluating models for re-ranking against a baseline while the second part exploits the performance of the ranking models in comparison to that of the experiments of other participants of TREC12 microblog.

## 6.1 Experiments on re-ranking models

There are two re-ranking models proposed. Model1 uses features extracted simply from the tweets and advocates using simplistic methodologies for re-ranking to improve efficiency, as opposed to adding extra features. Model2 adds the previously mentioned features along with scores from a run using BM25 incorporated into the re-ranking function.

Table 6.1 shows precisions for the top documents of the re-ranking function in comparison to the baseline model. For the baseline model a BM25 (b=075) weighting func-

Table 6.1: The precision comparison for top documents

| | **model1** | **model2** | **baseline** |
|---|---|---|---|
| **P@5** | 0.0628 | 0.0535 | 0.0535 |
| **P@10** | 0.0904 | 0.0927 | 0.0920 |
| **P@15** | 0.1068 | 0.1080 | 0.1056 |
| **P@20** | 0.1080 | 0.1080 | 0.1080 |

tion was used and Rocchios feedback algorithm for query expansion was applied. The left column shows the number of documents at which the precision was calculated. For example, p@5 indicates the precision for the top 5 retrieved documents. The numbers in the table show the calculated precisions of retrieving reached by models 1 and 2 and the baseline model respectively. As it is seen, for the first three rows of the table either model 1 or model 2 are outperformed the baseline model and in the last row the base model is not better. So we may conclude here that from the point of view of the precision our proposed model has a better performance. For the case of precision at 20 documents and more, the results are suffering from lack of data.

We then took a step forward in the analysis of model 1; i.e. the model without additional BM25. Figure 6.1 depicts how model1 performs in the top 10 results in comparison to the baseline. The vertical axis shows the number of queries in which the superiority of each of the runs is being studied, and the horizontal axis shows the number of first n hits. First ”n hits” in fact represents the n first number of the documents retrieved. The red bars indicate the number of queries where the base model performed better and showed a higher precision. Whereas, the green bars show the instances with higher performance of

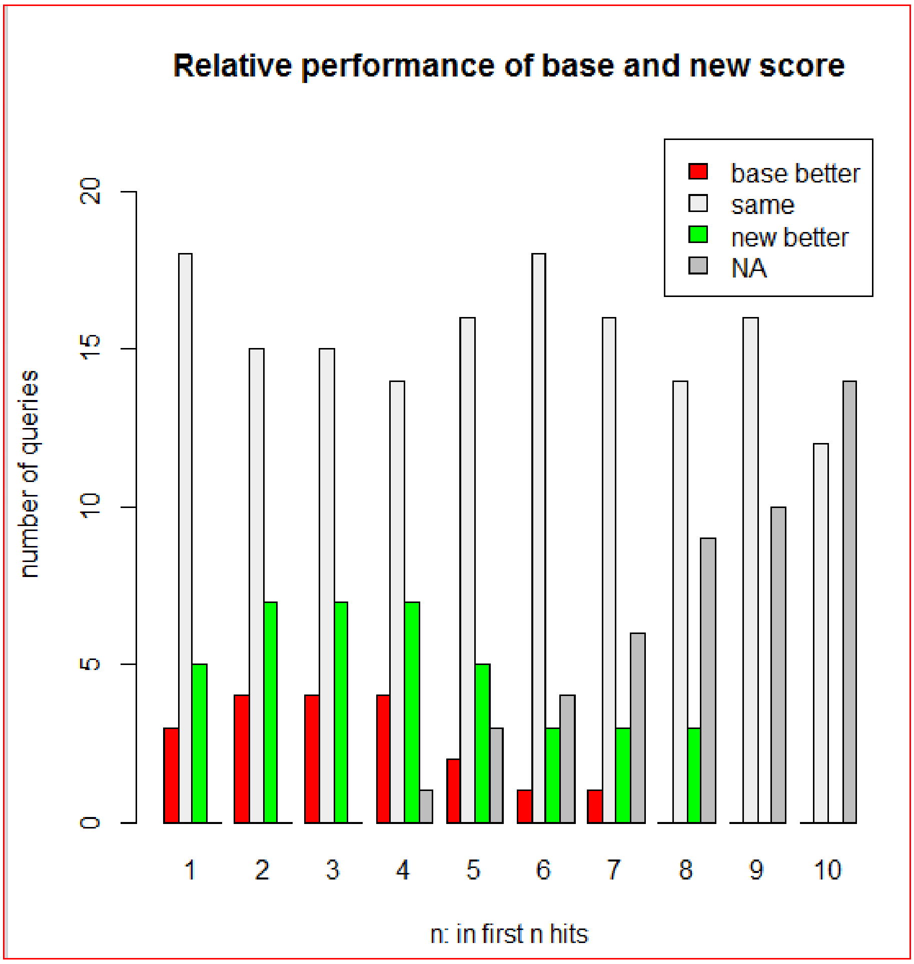


Figure 6.1: The relative performance of re-ranking models vs. the baseline

the first proposed model (model 1). The light gray bars are indicators of similar performances of both models and the darker gray bars point to unavailability of data. From this figure, it is evident that in all cases and for larger number of queries model1 is superior to the baseline. In the last instances, the missing data restricts the predictability of the performances; however, the overall trend denotes superiority of model 1.

## 6.2 Experiments on hybrid models

There are four different runs compared in this model.

- **YORK1:** A weighted Rocchio feedback model was used. The DFRee weighting model and the KL weighting model (doc=20, term=30, beta=1.4) for query expansion were applied.
- **YORK2:** Again a weighted Rocchio feedback model was used; The weighting model used was BM25 (b=0.3) and for query expansion KL weighting model (doc=20 term=30 beta=1.4) was applied.
- **york12mb3:** We use a weighted Rocchio feedback model, in which the DFRee weighting model and the KL weighting model (doc=20 term=30 beta=1) for query expansion were used. After that we conducted filtering according whether the tweet has links and hashtags.
- **york12mb4:**

  We use an enhanced Rocchio feedback model, in which the DFRee weighting model, the proximity model (weight=0.1 + FD + wins=8) and the KL weighting model (doc=20 term=30 beta=1) for query expansion were used.

### 6.2.1 Experimental Results

Different submitted runs show different precision on each topic. In order to investigate the similarity of the four runs and comparing them to median results a mean analysis is performed.

Figure 6.2, shows the mean comparison of York runs and median results. As it can be seen, the mean average precision of york12mb3 run is better than other runs; namely york2, york12mb4, and Median, but it is not evident that it is better than York1, due to the overlap in the confidence intervals. From this figure, we can categorize york1 and york12mb3 in one category which is superior to the other category consisting York2, york12mb4, and Median. Although, in all of four runs, we should expect similar results, the mean average precision of the median results is less than all of our four runs. A closer look to this figure can reveal the fact that York2 and york12mb4 are acting as if they are the same. So we may take these two out of our comparison and just compare York 1, york12mb3, and Median. The Median seems to be performing poorly in comparison to York 1, york12mb3 runs.

In Figure 6.3 the x-axis is the topic number and the y-axis is the average precision. In this figure, the comparison is made between the average precision of york12mb3 run and the median of all submitted runs to Microblog track 2012. The average precision for both of the runs is studied between all of the different queries and only a representative

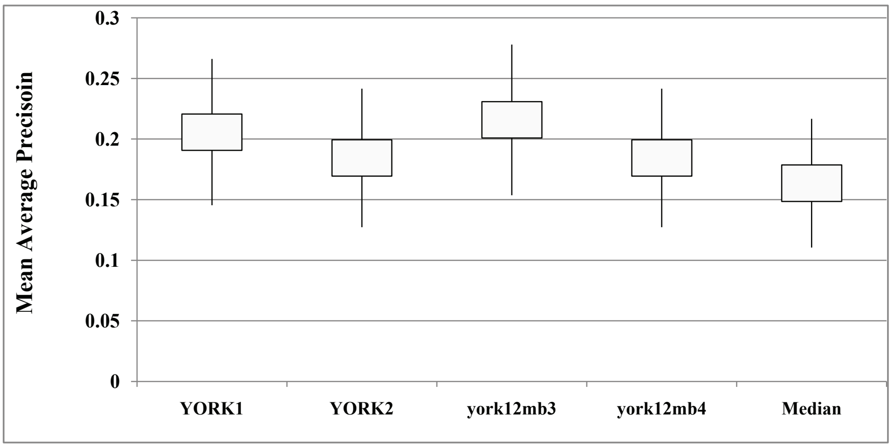


Figure 6.2: The mean comparison of York runs and median results

subsection is shown in this figure. The rationale behind studying the performance of the model between queries is based on different semantic nature of the queries. When average precision is being investigated, the effect of rare instances fades in the shadow of the more dominant effects. Figure 6.3 shows that in all of the topics except for topics71-73, york12mb3 shows higher precision values.

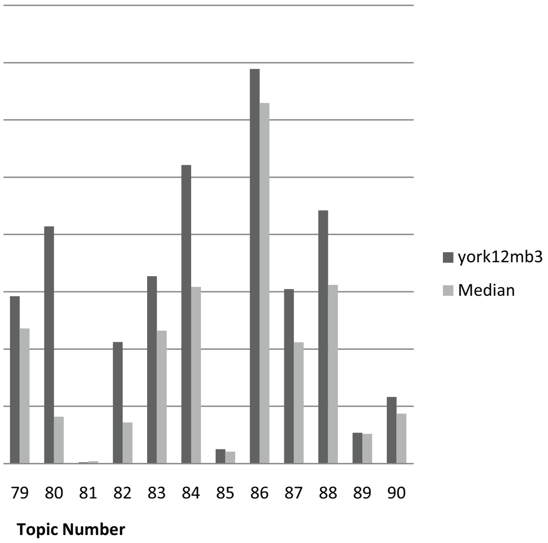


Figure 6.3: The average precision comparison of york12mb3 and median results

# 7 Conclusions

In the present work, we used logistic mixed models to analyze common features of Twitter to evaluate their predictive power of relatedness to the query. While some common textual and some tweet-specific features are included in the models, the focus is on exploiting the temporal patterns in the tweets and specifically how these patterns vary within different queries. We make use of the relevance judgments provided in the tweet11 corpus, which is the gold standard for evaluating retrieval results. To the best of our knowledge no other work has investigated the features for Twitter short text, using statistical analysis on a data set evaluated by experts, thus creating a gold standard. Moreover, the results of the analysis were applied to two different re-ranking models developed using generalized linear mixed effect models. In the course of the study , the low percentage of the number of relevant tweets for each query limited the power to detect effects and patterns. If this percentage were higher to 50%, our ability to predict would be improved. We would need to use considerably more than 1000 tweets per query. However the cost would be much higher. The standard derivation of estimated effect is roughly proportional to the square root of expected counts. Thus, we may reduce the standard errors of estimated effects to approximately a half, by increasing the number of tweets per query to 4000. Since we have selected the models through exploratory analyses, the results need to be interpreted with caution. Obtaining a model through an exploratory process can bias the achieved p-values. The results however are highly significant but it is worth mentioning that some of these variables were chosen before exploring the data and then interactions of variables with significant main effects were explored. Variables were picked based on the previous work in the literature and the focus was to include internal evidence than can be derived from the data set itself. These state-of-the-art variables can be listed as: the number of hashtags (hashTagCount), the number of mentions (mentionCount), the binary variable showing existence or non-existence of URLs (URLExist), the number of characters in the tweet (tweetLength), average term length (averageLength), as the variables driven from tweets; and number of terms (Number.of.Terms), and average length of terms (Average.term.length) of the query; other variables use information from both the tweet and the query. The tweet-query level variables studied in the present work are: the number of terms that occur both in the query and in the tweet (numOccur), the number of terms that occur in the query and in the hashtag of a tweet (numOccurHashtag), and the temporal variables (day and timeDif.s). Moreover, the interactions were picked out of all possible interactions; so these variables could be the lucky ones that happen to be significant.

As a future study it is important to identify what characteristics of queries are related to the time patterns. This could give us a source of additional information. The overall pattern of relevance shows a very strong temporal relationship that suggest that we can exploit patterns in time. It seems appropriate to classify the queries according to time sensitivity of relevance for that query. This would imply using semantic properties of the query since it is likely that the temporal patterns depend on the semantic characteristics of the query. In other words temporal information would require shifting to a semantic model for relevance.

This work could also be extended as follows: Features that fall into new categories such as features related to social networks and to semantic content have not been studied. This calls for a broader analysis that includes other aspects of microblog data set. In a previous research using a mention network[13], some of the social network aspects were taken into account and the results showed that a discussion of a topic creates a temporal cluster of people as opposed to a permanent social structure and to mentioning a particular individual does not necessarily make the social tie important for ad-hoc retrieval. The mention network is sensitive to the timeliness of the topics and unless the time feature is not included it would not create a powerful source for ad-hoc information retrieval. So in future work, including the social mention along with temporal aspects of the tweets could lead to new findings. Moreover, with the exception of using WordNet for expanding the

[13] A network of people where the relations form based on users mentioning each other or replying to the other person's tweet. See Appendix C

query topics, the main emphasis of this work was on the features derived from internal evidence, i.e. features that have been obtained from within the corpus. In contrast, TREC guidelines encourage authors to take advantage of external and future evidence. Hence, exploring features derived from external evidence seems like a valid next step to this work. We also wish to make use of the results of the analysis and to create a weighting function tailored specifically for the microblog dataset. In the second part of the study a ranking function and the results of our participation at TREC 2012 Microblog Track are provided. This part is mainly focused on evaluating a recently proposed hybrid retrieval model which has shown to be very effective on a large number of TREC datasets for ad-hoc information retrieval. The hybrid model is an extension of Rocchio's feedback method and incorporates three kinds of IR techniques, namely proximity, feedback document quality estimation, and query performance prediction techniques, under the pseudo relevance feedback (PRF) framework. In our experiments, we test different settings of this hybrid model on the microblog dataset. In two of the settings we used a modified BM25 specifically tailored for a Twitter dataset. Comparing the results to the submitted runs of Microblog track suggests relatively satisfactory results. We plan to make use of external evidence to improve the precision and also extend the weighting model using a larger training set.

# A R Code

## A.1 Explore

```
##
##  explore.R
##  Starting to explore models using split half
# Note that raw data
# were first save as .csv file
# Also the 'data' directory contains the query ids for the
#  learning data set.
library(spida)
library(lme4)
###############    run data.R to recreate data if necessary
data(dd)  # split half training data set
tab( dd, ~ id)
```

```
round(tab( dd, ~ id + relevancy, pct = 1),1)
round(tab( dd, ~ id + I(relevancy>0), pct = 1),1)

head(dd)
summary(dd)
xqplot( dd$timeDif.s./(3600*24))
xqplot( dd$numOccur)
xqplot( dd$numOccurHashTag)
xqplot( dd$hashTagCount)
# Variables in data
tab( dd, ~ numOccurHashTag )   # num of relevant hashtags
tab( dd, ~ hashTagCount)   # number of hashtags
# Derived variables
dd$hashtag.rel <- 1*(dd$numOccurHashTag > 0)
#at least one relevant hashtag
dd$hashtag.irrel <- 1*((dd$hashTagCount
- dd$numOccurHashTag) > 0)
# at least one irrelevant hashtab
dd$hashtag.extra <- dd$hashTagCount - dd$numOccurHashTag
# number of irrelevant hashtags
dd$timeDif.s. [dd$timeDif.s.==-1] <- NA  # -1 means NA
```

```
dd$day <- dd$timeDif.s. / (24*3600)      # in days
dd$num0 <- 1*(dd$numOccur == 1)
# rel is relevancy for logistic regression
dd$rel <- with( dd, ifelse( relevancy > 0, 1,
ifelse( relevancy == 0, 0, NA)))
tab( dd, ˜ rel + relevancy)
sp <- function(x) gsp( x, c(0, 1, 2) , 1, 0)
fit <- glmer ( rel ˜ hashtag.rel + URLExist + sp(day)
+ (1|id), dd,na.action = na.omit, family = binomial)
summary(fit)
wald(fit,’day’)
summary(dd)
pred <- expand.grid( day=seq(-2, 12, .1),
                     URLExist = 0:1,
                      hashtag.rel = 0:1)
pred$rel.1 <- predict( fit, pred, form = ˜ hashtag.rel + URLExist
+ sp(day))
pred$rel.p1 <- with(pred, 1/(1+exp(-rel.1)))
xyplot( rel.p1 ˜ day|URLExist , pred, groups = hashtag.rel,
 type = ’l’,
        auto.key = T, ylim = c(0, .2))
```

```
Cos <- function(x) {
      cbind( cos(2*pi*x), sin(2*pi*x), cos(4*pi*x) , sin(4*pi*x))
}
fit2 <- update( fit, . ˜. + Cos(day)*sp(day))
summary(fit2)
formula(fit2)
wald(fit2, ":")
wald(fit2, "Cos")
wald(fit2, "3$|4$")
wald(fit2, ":.*3$|:.*4$")
# log odds:
pred$rel.2 <- predict( fit2, pred, form =˜ hashtag.rel + URLExist
+ sp(day) + Cos(day) + sp(day):Cos(day))
# probability
pred$rel.p2 <- with(pred, 1/(1+exp(-rel.2)))
library(latticeExtra)
xyplot( rel.p1 ˜ day|URLExist , pred, groups = hashtag.rel,
type = ’l’, auto.key = T, ylim = c(0, .2),
        rel.p2 = pred$rel.p2, subscripts = TRUE) +
  glayer( panel.xyplot( y = rel.p2[subscripts],..., type = ’l’))
```

```
##

head(dd)

tab( dd, ~ is.na(BM25.194))

#####

#####   Looking at whether content of tweet add to relevance

#####


xqplot( dd$BM25.194)

fit <- glmer ( rel ~ hashtag.rel + URLExist + BM25.194 + (1|id),

               data =  subset(dd,

                              averageLength < 10 &

                                tweetLength < 141 &

                                averageLength > 3 ) ,

               na.action = na.omit, family = binomial)

summary(fit)
```

## A.2 Explore model

```
library(spida)

library(lme4)

if(FALSE){

# install some required packages and their dependencies
```

```
install.packages(c('car', 'rgl'))
# to install 'p3d'
download.file("http://blackwell.math.yorku.ca/R/p3d.zip",
 "p3d.zip")
install.packages("p3d.zip", repos = NULL)
}
library(p3d)
getFix.glmerMod <- function (fit, ...)
{
  ret <- list()
  ret$fixed <- fixef(fit)
  ret$vcov <- as.matrix(vcov(fit))
  ret$df <- rep(Inf, length(ret$fixed))
  ret
}
###############    run data.R to recreate data if necessary
data(dd)  # split half training data set
dim(dd)

ddsub <- subset(dd,
                averageLength < 10 &
```

```
                  tweetLength < 141 &
                  averageLength > 3 &
                  day >= 0)
###############################Two Models
############################# Final Models
sp <- function(x) gsp(x , c(1,6,13), 1, 0)
spa <-  function(x) gsp(x , c(4,5), 1, 0)
fitWOBM <- glmer ( rel ˜ tweetLength + hashTagCount + URLExist
                  +mentionCount  + spa(averageLength) +
                  sp(day)*numOccur+ sp(day)*
numOccurHashTag+(1|id),
                  data =  ddsub ,
                  na.action = na.omit, family = binomial)
summary(fitWOBM)
knot3 <- c(1,3,8)
sp3 <- function(x) gsp(x , knot3, 1, 0)

fitBM <- glmer(rel ˜ tweetLength + URLExist + BM25.194 +
 mentionCount + sp(day) + sp3(hashTagCount)  + (1 |id),
         data =  ddsub ,
         na.action = na.omit, family = binomial)
```

```
summary(fitBM)
#############Exploring models#############
fit <- glmer ( rel ~ tweetLength + hashTagCount+ hashtag.rel
 + URLExist + BM25.194 +mentionCount+ day +averageLength
 + numOccur +numOccurHashTag +(1|id), data =  ddsub ,
                 na.action = na.omit, family = binomial)
summary(fit)

day.knot <- c(1,6,13)#rounded 3tiles,
#for more convenience in interpretation
sp <- function(x) gsp(x , day.knot, 1, 0)
spa <-  function(x) gsp(x , c(4,5), 1, 0)
fit1 <- update(fit,.~. -day +sp(day))
summary(fit1)
##############Models without BM25###############
fit2 <- update(fit1, .~. -BM25.194)
summary(fit2)
fit3 <- update(fit2, .~. -sp(day) - numOccur+
sp(day)*numOccur)
summary(fit3)
```

```
wald(fit3,"HashTag")
fit3.n <- update(fit3, .˜. -numOccurHashTag)
summary(fit3.n)
fit4 <-  update(fit3, .˜. - numOccurHashTag +
numOccurHashTag*sp(day))
summary(fit4)
pred <- expand.grid( day=seq(0, 15, .1),
                     URLExist = 1,
                     averageLength = 5,
                     hashtag.rel = 1,
                     tweetLength=5 ,
                     hashTagCount=1 ,
                     mentionCount=3,
                     numOccur=2)
pred$rel.1 <- predict( fit4, pred,  REform = NA)
pred$rel.p1 <- with(pred, 1/(1+exp(-rel.1)))
xyplot( rel.p1 ˜ day , pred,  type = 'l',
        auto.key = T, ylim = c(0, .2))
pred <- expand.grid( day=seq(0, 15, .1),
                     URLExist = 1,
                     numOccurHashTag = 1,
                    averageLength = 5,
                    hashtag.rel = 1,
                    tweetLength=5 ,
                    hashTagCount=1 ,
                    BM25.194=25,
                    mentionCount=3,
                    numOccur=2)
pred$rel.2 <- predict( fit3, pred, , REform = NA)
pred$rel.p2 <- with(pred, 1/(1+exp(-rel.2)))
xyplot( rel.p2 ˜ day , pred, type = ’l’,
        auto.key = T, ylim = c(0, .25))
############################################
############building a model with BM25#########
############################################
wald(fit1, "hashtag.rel")
fit2BM <- update(fit1, .˜. -hashtag.rel)
summary(fit2BM)
wald(fit2BM, "Hash")
fit3BM <- update(fit2BM, .˜.- numOccurHashTag)
summary(fit3BM)
wald(fit3BM, "aver")
```

```
fit4BM <- update(fit3BM, .˜.- averageLength)
summary(fit4BM)
fit5BM <- update(fit4BM, .˜. -sp(day)-numOccur +
sp(day)*numOccur)
summary(fit5BM)
fit6BM <- update(fit4BM ,.˜. -numOccur)
summary(fit6BM)
pred <- expand.grid( day=seq(0, 17, .1),
                     URLExist = 1,
                     tweetLength=5 ,
                     hashTagCount=1 ,
                     BM25.194=25,
                     mentionCount=3,
                     numOccur=2)
pred$rel.3 <- predict( fit6BM, pred,  REform = NA)
pred$rel.p3 <- with(pred, 1/(1+exp(-rel.3)))
xyplot( rel.p3 ˜ day , pred, type = ’l’,
        auto.key = T, ylim = c(0, .35))
pred <- expand.grid( day=seq(0, 15, .2),
                     URLExist = 1,
                     tweetLength=5 ,
                    hashTagCount=1 ,
                    BM25.194= seq(14,25,1),
                    mentionCount=.3,
                    numOccur=2)
pred$rel3.1 <- predict( fit6BM, newdata = pred,
 REform = NA)
library(p3d)
Plot3d( rel3.1 ~BM25.194 + day, pred)
Axes3d()
day.knot2 <- c(1,3,7,13)
sp2 <- function(x) gsp(x , day.knot2, 1, 0)

fit7BM.sp2 <- update(fit6BM, .~. -sp(day) + sp2(day))
summary(fit7BM.sp2)

pred <- expand.grid( day=seq(0, 17, .1),
                    URLExist = 1,
                    tweetLength=5 ,
                    hashTagCount=1 ,
                    BM25.194=25,
                    mentionCount=3,
                     numOccur=2)

pred$rel.3 <- predict( fit7BM.sp2, pred,  REform = NA)
pred$rel.p3 <- with(pred, 1/(1+exp(-rel.3)))

xyplot( rel.p3 ˜ day , pred, type = ’l’,
        auto.key = T, ylim = c(0, .35))
fit7BM <- update(fit6BM, .˜. -sp(day)-URLExist +
sp(day)*URLExist)
summary(fit7BM)
tab(ddsub$hashTagCount )
day.knot3 <- c(1,3,8)
sp3 <- function(x) gsp(x , day.knot3, 1, 0)
fit9BM.sp3 <-update(fit6BM, .˜. - hashTagCount +
sp3(hashTagCount ))
summary(fit9BM.sp3)
L <- rbind(
"hashTagCount effect 0 to 1" = c(0,0,0,0,0, 0,0,0,0 ,1,0,0,0),
"hashTagCount effect 1 to 3" = c(0,0,0,0,0, 0,0,0,0 ,1,1,0,0),
"hashTagCount effect 3 to 8" = c(0,0,0,0,0, 0,0,0,0 ,1,1,1,0),
"hashTagCount effect 8+"     = c(0,0,0,0,0, 0,0,0,0 ,1,1,1,1)
```

```
  )
wald(fit9BM.sp3,L)


day.knot4 <- c(74,109,136)
sp4 <- function(x) gsp(x , day.knot4, 1, 0)


fit9BM.sp4 <-update(fit6BM, .˜. - tweetLength+ sp4(tweetLength))
summary(fit9BM.sp4)


L <- rbind(
"tweetLength effect 0 to 74"   = c(0,0,0,0,0, 0,0,0,0 ,1,0,0,0),
"tweetLength effect 74 to 109" = c(0,0,0,0,0, 0,0,0,0 ,1,1,0,0),
"tweetLength effect 109 to 136"= c(0,0,0,0,0, 0,0,0,0 ,1,1,1,0),
"tweetLength effect 136+"      = c(0,0,0,0,0, 0,0,0,0 ,1,1,1,1))


wald(fit9BM.sp4,L)
####################
fit1 <- update(fit, .˜ .-  numOccurHashTag)
summary(fit1)
wald(fit1,"hash")
fit2 <- update(fit1, .˜ .- hashtag.rel)
```

```
summary(fit2)

#

 fit3 <- update(fit2,.˜. -day +spexp(day))

 summary(fit3)

spa2 <- function(x) gsp(x, c(4,5), 1,0)

fit5 <- update(fit3, .˜ . + numOccurHashTag -

 averageLength+ spa2(averageLength))

summary(fit5)

L <- rbind(

"time effect 0 to 1.41"   = c(0,0,0,0, 0,0,0,0 ,1,0,0,0, 0,0,0,0),

"time effect 1.41 to 6.35"= c(0,0,0,0, 0,0,0,0 ,1,1,0,0, 0,0,0,0),

"time effect 6.35 to 13.5"= c(0,0,0,0, 0,0,0,0 ,1,1,1,0, 0,0,0,0),

"diff in slope at 6.35"   = c(0,0,0,0, 0,0,0,0 ,0,0,1,0, 0,0,0,0),

"time effect 13.5+"       = c(0,0,0,0, 0,0,0,0 ,1,1,1,1, 0,0,0,0),

"aveLen effect 0 to 4"    = c(0,0,0,0, 0,0,0,0,  0,0,0,0, 1,0,0,0),

"aveLen effect 4 to 5"    = c(0,0,0,0, 0,0,0,0,  0,0,0,0, 1,1,0,0),

"diff in slope at 4"      = c(0,0,0,0, 0,0,0,0,  0,0,0,0, 0,1,0,0))

wald(fit5,L)

pred <- expand.grid( day=seq(0, 12, .1),

                     URLExist = 1,

                     numOccurHashTag = 1,
                     averageLength = 5,

                     hashtag.rel = 1,

                     tweetLength=5 ,

                     hashTagCount=1 ,

                     BM25.194=25,

                     mentionCount=3,

                     numOccur=2,

                     numOccurDic=2

                     )

pred$rel.1 <- predict( fit5, pred, , REform = NA)


pred$rel.p1 <- with(pred, 1/(1+exp(-rel.1)))

xyplot( rel.p1 ˜ day|URLExist , pred, groups = hashtag.rel,

type = ’l’,auto.key = T, ylim = c(0, .2))

pred$rel.lo <- predict( fit3g2.q3, newdata = pred,

 REform = NA)

lo2p <- function(x) 1/(1+exp(-x))

lo2p(c(-3, -5))

probs <- c( .005,.007, .01,.015, .02,.025,.03, .04, .05)

p2lo <- function(x) log( x / (1-x))

library(latticeExtra)
```

```
xyplot( rel.lo ~ days, pred,
        ylab = 'probability of relevant tweet',
        xlab = 'tweet length',
        type = 'l',
        scales = list( y=list( at=p2lo(probs), labels =
                                paste( 100*probs,"%"))))+
  layer( panel.abline( h = p2lo(probs), col = 'gray'))
library(p3d)
Plot3d( rel.lo ~ day + averageLength, pred)
Axes3d()
fit6AV <- update(fit5, .~ .- BM25.194)
summary(fit6AV)
wald(fit6AV,L)
fit7 <- update(fit6AV,.~.-spexp(day) - numOccur
- numOccurDic+spexp(day)*numOccur)
summary(fit7)
fit8 <- update(fit7 ,.~.+  spexp(day)*numOccurHashTag)
summary(fit8)
```

## A.3 Descriptive

```
####################  Descriptive Statistics
```

```
library(spida)

library(lme4)

data(dd)

dd$timeDif.s. [dd$timeDif.s.==-1] <- NA  # -1 means NA

dd$day <- dd$timeDif.s. / (24*3600)      # in days

dd$num0 <- 1*(dd$numOccur == 1)

dd$rel <- with( dd, ifelse( relevancy > 0, 1,

ifelse( relevancy == 0, 0, NA)))

ddsub <- subset(dd,

                averageLength < 10 &

                  tweetLength < 141 &

                  averageLength > 3 &

                  day >= 0)

names(ddsub)

ddsub2<- subset(ddsub, rel>0)

tab( ddsub, ~ rel)

tab( ddsub, ~ numOccurHashTag)

tab( ddsub, ~ rel + numOccurHashTag)

tab( ddsub, ~ hashTagCount)

tab( ddsub, ~ rel + hashTagCount)

tab( ddsub, ~ numOccur)
tab( ddsub, ˜ rel +numOccur)
tab( ddsub, ˜ mentionCount)
tab( ddsub, ˜ rel +mentionCount)
tab( ddsub, ˜ URLExist)
tab( ddsub, ˜ rel + URLExist)
hist(ddsub$BM25.194, breaks=30, col="lightgrey",
main="Histogram of BM25 score", xlab="BM25")
hist(ddsub$tweetLength, breaks=50, col="lightgrey",
 main="Histogram of tweet length",
 xlab="tweet length")
par(mfrow = c(1,1) , pty = "s")
hist(ddsub$mentionCount, breaks=30, col="lightgrey",
 xlab="mentionCount",main=NULL)
hist(ddsub$hashTagCount, breaks=50, col="lightgrey",
 xlab="hashTagCount",main=NULL)
hist(ddsub$numOccur, breaks=50, col="lightgrey",
 xlab="numOccur",main=NULL)
# Some Plots
library(latticeExtra)
xyplot( jitter(rel) ˜ day, ddsub, pch = ’.’) +
layer( panel.loess(..., family = ’gaussian’,
 col = ’red’,lwd = 2))
xyplot( jitter(rel,.1) ~ day|factor(id), ddsub, pch =’.’) +
layer( panel.loess(..., family = ’gaussian’,
 col = ’red’,lwd = 2))
tab(ddsub$numOccurHashTag)
ddsub$rel.ave <- with(ddsub, capply( rel, id, mean,
 na.rm = T))
ddsub.id <- up(ddsub, ~ id)
ddsub$ido <- reorder( factor(ddsub$id), ddsub$rel.ave)
xyplot( jitter(rel,.1) ~ day|factor(ido), ddsub, pch =’.’,
ylab = "qrel score (with jitter)") +
  layer( panel.smoother(...,  col = ’red’,lwd = 2))

xyplot( hashTagCount ~ day|URLExist, ddsub2) +
layer( panel.loess(..., family = ’gaussian’,
col = ’red’,lwd = 2))
xyplot( jitter(rel)~  day| mentionCount, ddsub2, groups=URLExist)+
layer( panel.loess(..., family = ’gaussian’,
col = ’red’,lwd = 2))
fit.BM <- glmer ( rel ~  BM25.194 + (1|id),
                  data =  ddsub ,
                 na.action = na.exclude, family = binomial)
class(fit.BM)
showMethods( class=’glmerMod’)
zz <-residuals(fit.BM, type = ’working’)


ddsub$rel.BM25.wresid <- residuals(fit.BM, type = ’working’)
ddsub2 <- subset(ddsub, !is.na(rel.BM25.wresid))
ddsub2$ido2 <- with(ddsub2, reorder( factor(id), rel.ave))
xyplot( jitter(rel.BM25.wresid,.1) ˜
day|factor(ido2), ddsub2, pch =’.’,
 ylab ="working residual of qrel score predicted with BM25",
 ylim = c(-2,10))+ layer( panel.smoother(...,
col = ’red’,lwd = 2))
xyplot( jitter(rel.BM25.wresid,.1) ˜
day, ddsub2, groups = factor(ido2),
 pch =’.’,ylab = "working residual of qrel
score predicted with BM25",
 ylim = c(-2,10)) + glayer( panel.smoother(...,
 col = ’red’,lwd = 2))
## Summary of Some of the variables
summary(ddsub$rel)
summary(ddsub$tweetLength)
summary(ddsub$averageLength)
summary(ddsub$BM25.194)
summary(ddsub$hashTagCount)
summary(ddsub$mentionCount)
summary(ddsub$numOccur)
xyplot( numOccurHashTag ~ day|factor(id),ddsub, pch ='.') +
  layer( panel.loess(..., family = 'gaussian',
col = 'red',lwd = 2))
xyplot( numOccurHashTag ~ day|factor(id),
subset(ddsub,ddsub$numOccurHashTag<3), pch ='.') +
  layer( panel.loess(..., family = 'gaussian',
col = 'red',lwd = 2))
xyplot( numOccur ~ day|factor(id),ddsub, pch ='.') +
  layer( panel.loess(..., family = 'gaussian',
col = 'red',lwd = 2))
xyplot( numOccur ~ day|factor(id),
subset(ddsub,ddsub$numOccur<8), pch ='.') +
  layer( panel.loess(..., family = 'gaussian',
col = 'red',lwd = 2))
xyplot( numOccur ~ numOccurHashTag|factor(id),ddsub, pch ='.') +
  layer( panel.loess(..., family = ’gaussian’,
col = ’red’,lwd = 2))
xyplot( numOccur ˜ numOccurHashTag|factor(id),
subset(ddsub,ddsub$numOccurHashTag<3
&ddsub$numOccur<10), pch =’.’) +
  layer( panel.loess(..., family = ’gaussian’,
col = ’red’,lwd = 2))
xyplot( hashTagCount ˜ numOccurHashTag|factor(id),
subset(ddsub,ddsub$numOccurHashTag<4), pch =’.’) +
  layer( panel.loess(..., family = ’gaussian’,
col = ’red’,lwd = 2))
xyplot( BM25.194 ˜ day|URLExist,ddsub, pch =’.’) +
  layer( panel.loess(..., family = ’gaussian’,
col = ’red’,lwd = 2))
xyplot( relBM ˜ day|URLExist,gg, pch =’.’) +
  layer( panel.loess(..., family = ’gaussian’,
col = ’red’,lwd = 2))
xyplot( relWOBM ˜ day|URLExist,gg, pch =’.’) +
  layer( panel.loess(..., family = ’gaussian’,
col = ’red’,lwd = 2))
hist(ddsub$BM25.194, breaks=30, col="lightgrey",
 main="Histogram of BM25 score", xlab="BM25")
hist(ddsub$averageLength, breaks=50, col="lightgrey",
main="Histogram of average length of the terms in a query",
xlab="average length")
uid <-unique(ddsub$id)
par(mfrow = c(2, 3) , pty = "s")
for( i in 1:6){
av <-with(ddsub, subset(averageLength, id==uid[i]))
hist(av, breaks=50, col="lightgrey", xlab="average length")
}
for( i in 7:12){
av <-with(ddsub, subset(averageLength, id==uid[i]))
hist(av, breaks=50, col="lightgrey", xlab="average length")
}
for( i in 13:18){
av <-with(ddsub, subset(averageLength, id==uid[i]))
av <-with(ddsub, subset(averageLength, id==uid[i]))
hist(av, breaks=50, col="lightgrey", xlab="average length")
}
for( i in 19:24){
av <-with(ddsub, subset(averageLength, id==uid[i]))
```

```
hist(av, breaks=50, col="lightgrey", xlab="average length")
}
```

## A.4 Final Analysis

```
############################
###############################Some tests for the Two Models
sp <- function(x) gsp(x , c(1,6,13), 1, 0)
spa <-  function(x) gsp(x , c(4,5), 1, 0)

fitWOBM <- glmer ( rel ~ tweetLength + hashTagCount
+ URLExist +mentionCount  + spa(averageLength) +
                     sp(day)*numOccur+ sp(day)*
numOccurHashTag+(1|id),data = ddsub,
na.action = na.omit, family = binomial)
summary(fitWOBM)
knot3 <- c(1,3,8)
sp3 <- function(x) gsp(x , knot3, 1, 0)

fitBM <- glmer(rel ~ tweetLength + URLExist + BM25.194 +
 mentionCount +sp(day) + sp3(hashTagCount)  + (1 |id),
               data =  ddsub ,
               na.action = na.omit, family = binomial)
summary(fitBM)
################################TESTs
summary(fitWOBM)
wald(fitWOBM)  # not
wald(fitWOBM,-1) # but this the right test
wald(fitWOBM,":")  #overall test for interactions
wald(fitWOBM,":.*r$")  #test for interactions between day
#and numOccur
wald(fitWOBM,":.*g$")  #overall test for interactions
#  between day andnumoccurhashtag
wald(fitWOBM,"day")  #overall test for day
wald(fitWOBM,"day.*C")  # overall test for spline for day
       # i.e. did we need something
        #more complex than a linear effect of day
wald(fitWOBM,"numOccur$")
wald(fitWOBM,"numOccurHashTag")
wald(fitWOBM,"average")
wald(fitWOBM,"URLExist")
wald(fitWOBM,"tweetLength")
wald(fitWOBM,"mentionCount")
wald(fitWOBM,"hashTagCount")


wald(fitBM,-1)
wald(fitBM,":")
wald(fitBM,"day")
wald(fitBM,"hashTagCount")
wald(fitBM,"tweetLength")
wald(fitBM,"URLExist")
wald(fitBM,"BM25")
wald(fitBM,"mentionCount")
######################################
pred <- expand.grid( day=seq(0, 16, .1),
                     URLExist = 1,
                     numOccurHashTag = 1,
                     averageLength = 5,#c(0,12,0.5),
                     tweetLength=5 ,
                     hashTagCount=2,#c(0,8,1) ,
                     mentionCount=3,
                     numOccur=2)
pred2 <- expand.grid( day=3,#seq(0, 16, .1),
                      URLExist = 1,
                      numOccurHashTag = 1,
                      averageLength = 3,
                      tweetLength=5 ,
                      hashTagCount=c(0,8,1) ,
                      mentionCount=3,
                      numOccur=2)
pred$relWOBM <- predict( fitWOBM, pred,
 REform = NA)
pred$rel.p <- with(pred, 1/(1+exp(-relWOBM)))
xyplot( rel.p ˜ day , pred, type = ’l’,
        auto.key = T, ylim = c(0, .25))

pred2 <- expand.grid( day=seq(0, 16, .1),
                      URLExist = 1,
                      numOccurHashTag = 1,
                      averageLength = seq(0,12,0.5),
                      tweetLength=5 ,
                      hashTagCount=2 ,
                      mentionCount=3,
                      numOccur=2)
pred2$relWOBM2 <- predict( fitWOBM, pred2,
 REform = NA)
pred2$rel.p2 <- with(pred2, 1/(1+exp(-relWOBM2)))
library(p3d)
Plot3d( rel.p2 ˜ averageLength  + day, pred2)
Axes3d()
############################
################################
pred <- expand.grid( day=seq(0, 16, .1),
                     URLExist = 1,
                     numOccurHashTag = 1,
                     averageLength = 5,#c(0,12,0.5),
                     tweetLength=5 ,
                     hashTagCount=2,#c(0,8,1) ,
                     mentionCount=3,
                     numOccur=2)
pred2 <- expand.grid( day=3,#seq(0, 16, .1),
                     URLExist = 1,
                     numOccurHashTag = 1,
                     averageLength = 3,
                     tweetLength=5 ,
                     hashTagCount=c(0,8,1) ,
                     mentionCount=3,
                     numOccur=2)
pred$relWOBM <- predict( fitWOBM, pred,
 REform = NA)
pred$rel.p <- with(pred, 1/(1+exp(-relWOBM)))
xyplot( rel.p ˜ day , pred, type = ’l’,
        auto.key = T, ylim = c(0, .25))
summary(gg$averageLength)
pred2 <- expand.grid( day=seq(0, 16, .1),
                      URLExist = 1,
                      numOccurHashTag = 1,
                      averageLength = seq(0,12,0.5),
                      tweetLength=5 ,
                      hashTagCount=2 ,
                      mentionCount=3,
                      numOccur=2)
pred2$relWOBM2 <- predict( fitWOBM, pred2,
 REform = NA)
pred2$rel.p2 <- with(pred2, 1/(1+exp(-relWOBM2)))
library(p3d)
Plot3d( rel.p2 ˜ averageLength  + day, pred2)
```

```
Axes3d()

##################################R-Precision

data(dd)

dim(dd)

dd$rel[is.na(dd$rel)] <- 0

precision <- with(dd,tapply(rel,queryId,mean))
```

# B Java code for document extraction and analysis

## B.1 Main Class

```
import java.util.*;

import java.io.*;


public class MainClass {


public static void main(String [] args){

System.out.println("Start !");

Test1 t1 = new Test1();

Map<Integer,String> queryContentMap = t1.getQueryContent();


Map<Integer,Long> queryTimeMap = t1.getQueryTime();
```

```
ArrayList<Object> TQueryL = t1.getTweetQueryMaps();


Map<Integer, Map<String, Float>>

 tweetQueryMaps = (Map<Integer, Map<String, Float>>)

TQueryL.get(0);

Set<String> tweetIdSet = (Set<String>)TQueryL.get(1);


Map<String, Tweet> tweetMap = t1.getTweetMap(tweetIdSet);


try{

FileWriter fileWriter = new FileWriter("/Directory/Output.txt");

BufferedWriter bw = new BufferedWriter(fileWriter);


String line = "tweetId" + "\t" + "queryId" + "\t" +

"tweetLength" + "\t" + "relevancy" + "\t" +  "hashTagCount" +

"\t" + "URLExist" + "\t" + "mentionCount" +"\t" + "averageLength"

+ "\t" + "numOccur" + "\t" + "numOccurHashTag" + "\t" +

"timeDif(s)" +"\t" + "tweetTime(s)" +"\t" + "numOccurHashTagDic"+

"\t" + "content" + "\t";

 bw.write(line + "\n");
```

```
int counterTemp = 0;

int counterTemp2 = 0;


for(Iterator<Integer> iterator1 = tweetQueryMaps.keySet()

.iterator(); iterator1.hasNext();){

int queryId = iterator1.next();

String queryContent = queryContentMap.get(queryId);

long queryTime = queryTimeMap.get(queryId);


for(Iterator<String> iterator2 = tweetQueryMaps.get(queryId)

.keySet().iterator();

iterator2.hasNext();){

         String tweetId = iterator2.next();

         float relevancy = tweetQueryMaps.get(queryId).

get(tweetId);

         try{

            if(tweetMap.containsKey(tweetId)){

 Tweet tweet = tweetMap.get(tweetId);


 int hashTagCount = tweet.content.length() -

 tweet.content.replaceAll("#", "").length();
```

```
int URLExist = 0;

 if(tweet.content.contains("http://"))

URLExist = 1;


int mentionCount = tweet.content.length() -

 tweet.content.replaceAll("@", "").length();


int occurCount = getNumberOfOccurances(queryContent,

tweet.content);


int occurCountHashTag = getNumberOfOccurancesHashTag

(queryContent,

tweet.content);



 long timeDif = -1;

 long tweetTime = -1;

 if(tweet.time != null && !tweet.time.equals("null")){

tweetTime = Test1.convertToDate(tweet.time);

timeDif = queryTime - tweetTime;

timeDif = timeDif/1000;
tweetTime = tweetTime/1000;
          }

line = tweetId + "@" + "\t" + queryId + "\t" +
tweet.content.length() + "\t" +relevancy + "\t"+
hashTagCount + "\t" + URLExist + "\t"+mentionCount +
"\t" + getAverageTermLength(tweet.content) + "\t" +
 occurCount + "\t" +occurCountHashTag + "\t" + timeDif
 + "\t" + tweetTime + "\t" + occurCountHashTagDic +
"\t" + tweet.content + "\t";
bw.write(line + "\n");
System.out.println((counterTemp2 ++) + "***");
             }
 }catch(Exception ex2){
   System.out.println("2>>>> " + (counterTemp ++) + "*" +
tweetId + "*" + ex2.getMessage());  }
           }
       }
       bw.close();
}catch(Exception ex1){
System.out.println("1>>>> " + ex1.getMessage());
```

```
}

System.out.println("End !");

return;
}

public static int getNumberOfOccurancesHashTag(String query,
 String tweet){

int result = 0;
String qq[] = query.split(" ");
String tt[] = tweet.split(" ");

for(int i=0; i<qq.length; i ++){
for(int j=0; j<tt.length; j ++){
if(tt[j].startsWith("#")){
String newT = tt[j].substring(1, tt[j].length());
if(newT.toLowerCase().equals(qq[i].toLowerCase()))
result ++;
}
}

}


return result;

}


public static int getNumberOfOccurances(String query,

 String tweet){


int result = 0;

String qq[] = query.split(" ");


for(int i=0; i<qq.length; i ++){

result += countOcc(tweet, qq[i].toLowerCase());

}

return result;

}

public static int countOcc(String tweet, String s){

int result = 0;

String tt[] = tweet.split(" ");

for(int i=0; i<tt.length; i ++){
```

```
if(tt[i].toLowerCase().equals(s.toLowerCase()))
result ++;
}

return result;
}
public static double getAverageTermLength(String s){
int numberOfValidWords = 0;
int totalLength = 0;
String []ss = s.split(" ");
for(int i=0; i<ss.length; i ++){
String word = ss[i];
if(!word.startsWith("@")){
if(!word.contains("http://")){
numberOfValidWords ++;
totalLength += word.length();
if(word.startsWith("#"))
totalLength = totalLength - 1;
}
}
}
```

```
if(numberOfValidWords == 0)
return 0;
else{
double result = (double)totalLength/numberOfValidWords;
result = (double)Math.round(result * 100) / 100;
return result;
}
}
}
```

## B.2 Test1 Class

```
import java.io.*;
import java.util.*;

public class Test1 {
public Map<Integer,String> getQueryContent(){
//read the 60000 tweet ids
Map<Integer,String> queryTimeMap = new HashMap<Integer,
String>();
        try {
            BufferedReader input = new BufferedReader(new
FileReader(new File("/Directory/QueryContent.txt")));
            String line = null;
            int count = 0;
            while ( (line = input.readLine()) != null  ){
             count ++;
             line = line.trim();
             queryTimeMap.put(count, line);
            }

        }catch(Exception ex){
        }
        return queryTimeMap;
}

public Map<Integer,Long> getQueryTime(){
//read the 60000 tweet ids
Map<Integer,Long> queryTimeMap = new HashMap<Integer,
Long>();
        try {
            BufferedReader input = new BufferedReader
```

```
(new FileReader(new File("/Directory
/QueryTimes.txt")));
            String line = null;
            int count = 0;
            while ( (line = input.readLine()) != null  ){
             count ++;
             line = line.trim();
             long dt = convertToDate(line);
             queryTimeMap.put(count, dt);
            }

        }catch(Exception ex){
        }
        return queryTimeMap;
}

public static long convertToDate(String s){
//Date(int year, int month, int date, int hrs,
int min,int sec)
//"Fri Feb 04 14:36:59 +0000 2011"
s = s.trim();
```

```
String []ss = s.split(" ");
int year = Integer.parseInt(ss[5]) - 1900;

int month = 0;
if(ss[1].equals("Jan"))
month = 0;
if(ss[1].equals("Feb"))
month = 1;
if(ss[1].equals("Mar"))
month = 2;
if(ss[1].equals("Apr"))
month = 3;
if(ss[1].equals("May"))
month = 4;
if(ss[1].equals("Jun"))
month = 5;
if(ss[1].equals("Jul"))
month = 6;
if(ss[1].equals("Aug"))
month = 7;
if(ss[1].equals("Sep"))
```

```
month = 8;

if(ss[1].equals("Oct"))

month = 9;

if(ss[1].equals("Nov"))

month = 10;

if(ss[1].equals("Dec"))

month = 11;

int date = Integer.parseInt(ss[2]);

String []tt = ss[3].split(":");

int hrs = Integer.parseInt(tt[0]);

int min = Integer.parseInt(tt[1]);

int sec = Integer.parseInt(tt[2]);


//return new Date(year, month, date, hrs,

min, sec);

return Date.UTC(year, month, date, hrs, min, sec);

}


public ArrayList<Object> getTweetQueryMaps(){

//read the 60000 tweet ids

Map<Integer, Map<String, Float>> tweetQueryMaps =
new HashMap<Integer, Map<String, Float>>();
Set<String> tweetIdSet = new HashSet<String>();
        try {
            BufferedReader input = new BufferedReader(new
FileReader(new File("./TB")));
   String line = null;
     while ( (line = input.readLine()) != null  ){
           try{
String []ss = line.split(" ");
              String queryIdStr = ss[0];
              queryIdStr = queryIdStr.replace("\"", "");
              int queryId = Integer.parseInt(queryIdStr);
              String tweetId = ss[2];
              String tweetScoreStr = ss[3];
              tweetScoreStr = tweetScoreStr.replace("\"", "");
              float tweetScore = Float.parseFloat(tweetScoreStr);
              if(!tweetQueryMaps.containsKey(queryId)){
              tweetQueryMaps.put(queryId, new HashMap<String,
Float>());
                 }
                 tweetQueryMaps.get(queryId).put(tweetId,
 tweetScore);
                 tweetIdSet.add(tweetId);
                }catch(Exception e){
                    continue;
                }
            }
        }catch(Exception ex){
            System.out.println(ex);
        }
        ArrayList<Object> result = new ArrayList<Object>();
        result.add(tweetQueryMaps);
        result.add(tweetIdSet);
        return result;
}
public Map<String, Tweet> getTweetMap(Set<String> tweetIdSet){
        Map<String, Tweet> tweetMap = new HashMap<String,
Tweet>();
        try {
         BufferedReader input = new BufferedReader(new
FileReader(new File("/UDirectory/statistics")));
            String line = null;
```

```
        //test the writing

            String s = "", bline = null;

           while ((bline = input.readLine()) != null) {

               s += bline+ "\n";

               }

               System.out.print(s);

            input.close();

         input = new BufferedReader(new FileReader(new
File("/Directory/statistics")));

         FileWriter fileWriter = new
 FileWriter("/Directory/NasiTweetsNasi.txt");

         BufferedWriter bw = new BufferedWriter(fileWriter);

            while ( (line = input.readLine()) != null  ){


             try{

             int p1 = line.indexOf("<num>");

             int p2 = line.indexOf("</num>");

             String tweetId = line.substring(p1+5, p2);

             if(tweetIdSet.contains(tweetId)){

             p1 = line.indexOf("<status>");

             p2 = line.indexOf("</status>");
             int tweetStatus = Integer.parseInt
(line.substring(p1+8,
p2));
             p1 = line.indexOf("<querytime>");
             p2 = line.indexOf("</querytime>");
             String tweetTime = line.substring(p1+11, p2);
             p1 = line.indexOf("<content>");
             p2 = line.indexOf("</content>");
             String tweetContent = line.substring(p1+9,
p2);
             Tweet tweet = new Tweet(tweetId,
tweetStatus,tweetTime, tweetContent);
             tweetMap.put(tweetId, tweet);

             bw.write(line+"\n");
             System.out.println("reading");
             }
             }catch(Exception ex1){
             System.out.println("*" + ex1.getMessage());
             }
            }
            bw.close();

        }catch(Exception ex2){

         System.out.println(ex2.getMessage());

        }

        return tweetMap;

}

}
```

## B.3 Tweet Class

```
public class Tweet {

public static int STATUS_HIGHLY_RELEVANT = 2;

public static int STATUS_RELEVANT = 1;

public static int STATUS_NOT_RELEVANT = 0;

public static int STATUS_SPAM = -2;


public String id;

public int status;

public String time;

public String content;


public Tweet(String id, int status,
String time, String content){
this.id = id;
this.status = status;
this.time = time;
this.content = content;
}
}
```

# C Mention Network

Twitter is probably the world's most famous microblogging service. In Twitter, users are allowed to share 140-character posts called Tweets. What Twitter introduces to world is a network of people sharing information and thoughts. This platform raises lots of opportunities for research, since it has plenty of informative characteristics that can be used. In addition, Twitter has the characteristics of both blogs and social networks in the form of a short text.

## C.1 Retweets

In network-driven social media conversations that happen people follow conversations in the context of individuals has enabled conversations to occur asynchronously and beyond geographic constraints, but they are still typically bounded by a reasonably well-defined group of participants in some sort of shared social context. Network-driven genres (e.g., social network sites, microblogging) complicate this because people follow the conversations in the context of individuals, not topical threads. Yet, conversations still emerge between dyads and among groups. On Twitter, a popular microblogging service, directed conversations usually involve use of the ”@user” syntax to refer to others and address messages to them [7]

The turning point of Twitter being a social network introduces some opportunities to make up for the lack of textual features. propose that popular users who have high in-degree are not necessarily influential in terms of spawning retweets or mentions.

In the present work we are also aiming to use this hidden network. We are proposing a novel perspective for information retrieval. In other word, we analyze a network of people who interact by mentioning each other as opposed to merely following. This usually means either a reply, which is an update, posted by clicking the Reply button on a Tweet, or a mention that is any Twitter update that contains ”@username” anywhere in the body of the tweet. The third case would be in the instances where someone shares a post from another user to their network of followers, or ”retweet”s in the form of RT @user. We believe that this studying this network can enlighten the path for research in the information retrieval of Twitter network in order to get the most relevant information from the limited text features.

In this part of the study, some minor preprocessing is conducted to remove non-English, forbidden, and null tweets from the dataset first. The only information needed for building the network in tweet content is the users that have been mentioned or whom their tweets have been retweeted. Therefor other conventional text preprocessing steps are ignored, as opposed to the experiments explained in section in section III that were dealing with texts. From the dataset we extract the short text documents that contain a username followed by the @ symbol; this usually means either a reply or a mention. @Reply is any update posted by clicking the Reply button on a Tweet. These tweets begin with a username whose tweet is getting a reply. A mention on the other hand is any Twitter update that contains ”@username” anywhere in the tweet; hence a reply can also be considered as a mention. More than one user can be addressed in a mention tweet. There is third class of tweets in which the symbol @ and a username appears, which is in the case of retweet. When a user wants to repost an interesting/informative tweet of another user, they can copy the content and add ”RT @user”. Thus, the usernames are extracted from the body of the tweets and a directed social relation is defined from the user who has posted the tweet to the one receiving it. Out of about 16 million tweets, about 6 million, or about 40% are of such quality. If a user was mentioned more than once, the value of the arc weight is increased subsequently. However the number of the directed connections, or arcs is almost the same as the number of the nodes. Table C.1 lists a general list of information about this network. The values in the table can be interpreted as follows:

- Number of lines in the network: About 7% of the lines have a value greater than 1, meaning that only 7% of interacting users have been active for multiple times.
- Density: The network density measure indicates a very scattered network. This

Table C.1: Properties of Twitter Mention Network

| Network Property | value |
|---|---|
| Number of lines with value =1 | 5814556 |
| Number of lines with value >1 | 433173 |
| Number of Loops | 8841 |
| Density | 0.00000018 |
| Average Degree | 2.11113970 |

shows that most of the interacting users, have addressed inactive people in the two-weeks period of data collection. This network needs the information of both parties in order to satisfy more information needs.

- The average degree in this network is 2.11, which means that there is only one tie between each two actors.
- Loops: 8841 loops exist in the network. These loops could be considered as small communities of users interacting during the specified time window.

These values suggest that even though a significant number of users mention another user, not all the addressed users were active in the same time window. That is why not many ties are occurred in the existing network of two weeks.

## C.2 Network Analysis

For the constructed network of people mentioning and retweeting each other, or what we call ”The Mention Network”, a partition is made consisting of node information for clusters that each contain related tweets to a query in the corpus. A total number of 110 query topics are included in the dataset. The first 50 were designed for microblog11, and the rest for microblog12. Experts have chosen these topics based on news making events and popular topic in the time period that the tweets have been crawled. The relevancy information of each tweet to the queries is provided in the relevance judgment in the corpus. In our study, highly relevant and relevant tweets scored 2 and 1 respectively by the human assessors, are treated the same. All other tweets that do not belong to any of the topic clusters have been assigned to a cluster 0. The result is a partition consisting of 111 clusters. Table II lists the most populated clusters and their corresponding topics. It is not surprising to observe that 99.98% of the tweets belong to the cluster 0. Meaning that not many interactions occur between users with informative tweet contents. This leaves us with a very limited number of actual nodes in the network. A quick glance to table II gives us some information of the topics that have caused initiation of discussions. ”McDonalds Food” is by far the most occurred topic in the dataset. This indicates that it was a debatable topic among twitter users. A closer look in the local view of the cluster of this topic shows no interaction between the discussers. Topics like ”MacDonalds food” and ”Texting and Driving”, share a quality of being relevant for a long period of time. Twitter information retrieval is extremely time sensitive. It is really important to study each topic around the peak time of the broadcast of the relevant news.

Table C.2: Topics with highest frequency

| Query id | Topic | Frequency | Frequency % |
|---|---|---|---|
| 0 | Irrelevant/Spam | 5917984 | 99.9859 % |
| 78 | McDonalds Food | 107 | 0.0018 % |
| 99 | Super Bowl Commercials | 41 | 0.0007 % |
| 66 | Journalists' treatment in Egypt | 36 | 0.0006 % |
| 104 | Texting and driving | 35 | 0.0006 % |
| 72 | Kardashians opinions | 33 | 0.0006 % |

# Bibliography

[1] Y. Aboulnaga and C. L. A. Clarke. Universitas Indonasia at TREC2012 Microblog Track. In *Proc. TREC 2012 Microblog track*, 2012.

[2] G. Amati, G. Amodeo, M. Bianchi, A. Celi, C. D. Nicola, and M. Flammini. FUB, IASI-CNR, UNIVAQ at TREC 2011 Microblog track. In *Proc. TREC 2011 Microblog track*, 2011.

[3] H. Amiri, Y. Bao, A. Cui, A. Datta, F. Fang, and X. Xu. NUSIS at TREC 2011 Microblog Track: Refining query results with hashtags. In *Proc. of TREC'11*, 2011.

[4] A. Bandyopadhyay, M. Mitra, and P. Majumder. Indian Statistical Institute at TREC 2011 Microblog Track. In *Proc. of TREC'11*, 2011.

[5] G. Berardi, A. Esuli, D. Marcheggiani, and F. Sebastiani. ISTI at TREC Microblog Track 2011: Exploring the use of hashtag segmentation and text quality ranking. In *Proc. of TREC'11*, 2011.

[6] S. Bhattacharya, C. Harris, Y. Mejova, C. Yang, and P. Srinivasan. The University

of Iowa at TREC 2011: Microblogs, medical records, and crowdsourcing. In *Proc. of TREC'11*, 2011.

[7] D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on Twitter. In *HICSS*, 2010.

[8] M. Bron, E. Meij, M. Peetz, and M. Tsagkias. Team COMMIT at TREC 2011. In *Proc. of TREC'11*, 2011.

[9] M. Cha, H. Haddadi, F. Benevenuto, , and K. P. Gummadi. Measuring user influence in Twitter: The million follower fallacy. In *Proc. of intl. AAAI Conf. on Weblogs and Social*, 2010.

[10] H. Chen and D. Karger. Less is more: Probabilistic models for retrieving fewer relevant document. In *Proc. of SIGIR'06*, 2006.

[11] M. Efron. The University of Illinois' graduate school of library and information science at TREC 2011. In *Proc. of TREC'11*, 2011.

[12] J. Faraway. *Statistical Learning From a Regression Perspective*. Chapman Hall/CRC Press, 2006.

[13] P. Ferguson, N. O'Hare, J. Lanagan, and A. F. Smeaton. CLARITY at the TREC 2011 Microblog Track. In *Proc. TREC 2011 Microblog track*, 2011.

[14] Z. Han, X. Li, M. Yang, H. Qi, S. Li, and T. Zhao. HIT at TREC 2012 Microblog Track. In *Proc. TREC 2012 Microblog track*, 2012.

[15] F. E. Harrel. *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. Springer, 2001.

[16] B. He, J. X. Huang, and X. Zhou. Modeling term proximity for probabilistic information retrieval models. *Information Sciences*, 181:3017–3031, 2011.

[17] C. Honeycutt and S. C. Herring. Beyond microblogging: Conversation and collaboration via Twitter. In *Proceedings of the 42nd Hawaii International Conference on System Sciences*, 2009.

[18] D. Howes. On relation between the intelligibility and frequency of occurrence of english words. *The Journal of Acoustical Society of America*, 29, 1957.

[19] M. Ibrahim, C. Vania, F. Rahman, C. Atimas, and M. Adriani. Frequent itemset mining for query expansion in microblog ad-hoc search. In *Proc. TREC 2012 Microblog track*, 2012.

[20] A. Java, X. Song, T. Finin, and B. Tseng. Why We Twitter: Understanding microblogging usage and communities. In *Procedings of the Joint 9th WEBKDD and 1st SNA-KDD Workshop 2007*, 2007.

[21] Y. Jiang and C. Scott. Predicting the speed, scale, and range of information diffusion in Twitter. In *International Conference on Weblogs and Social Media*, 2010.

[22] J. Jmal and R. Faiz. Customer review summarization approach using Twitter and SentiWordNet. In *WIMS '13 Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*, 2013.

[23] R. Kelly. Twitter study: August 2009, August 2009.

[24] Y. Kim, R. Yeniterzi, and J. Callan. Overcoming vocabulary limitations in Twitter Microblogs. In *Proc. TREC 2012 Microblog track*, 2012.

[25] Y. Li, Z. Zhang, W. Lv, and Q. Xie. PRIS at TREC2011 Micro-blog Track. In *Proc. TREC 2011 Microblog track*, 2011.

[26] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[27] S. Louvan, M. Ibrahim, M. Adriani, C. Vania, B. Distiawan, and M. Z. Wanagiri. University of Indonesia at TREC 2011 Microblog Track. In *Proc. TREC 2011 Microblog track*, 2011.

[28] D. Metzler and C. Cai. USC/ISI at TREC 2011: Microblog Trr1ack. In *Proc. TREC 2011 Microblog track*, 2011.

[29] L. Mitchell, K. D. Harris, M. R. Frank, P. S. Dodds, and C. M. Danforth. The geography of happiness: Connecting Twitter sentiment and expression, demographics, and objective characteristics of place. In *The Computing Research Repository*, 2013.

[30] T. Miyanishi, N. Okamura, X. Liu, K. Seki, and K. Uehara. TREC 2011 microblog track experiments at Kobe University. In *Proc. of TREC'11*, 2011.

[31] T.-S. Moh and A. J. Murmann. Can you judge a man by his friends? enhancing spammer detection on the Twitter microblogging platform using friends and followers. *Information Systems, Technology and Management Communications in Computer and Information Science*, 54:210–220, 2010.

[32] M. Naaman, J. Boase, and C. Lai. Is it really about me? message content in social awareness streams. In *CSCW '10 Proceedings of the 2010 ACM conference on Computer supported cooperative work*, 2010.

[33] Z. Obukhovskaya, K. Pervyshev, A. Styskin, and P. Yandex. Yandex at TREC 2011 Microblog Track. In *Proc. of TREC'11*, 2011.

[34] I. Ounis, J. Lin, and I. Soboroff. Overview of the TREC-2011 microblog track. In *Proc. of TREC'11*, 2011.

[35] M. Petri, J. Shane, and F. Scholer. RMIT at TREC 2011 Microblog Track. In *Proc. of TREC'11*, 2011.

[36] J. C. Pinheiro and D. Bates. Springer, 2000.

[37] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei. Rumor has it: identifying misinformation in Microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011.

[38] J. Rocchio. Relevance feedback in information retrieval. In *Prentice-Hall Englewood Cliffs*, 1971.

[39] J. A. Rodriguez, A. J. Macminn, and J. M. Jose. University of Glasgow (uog-tw) at TREC Microblog 2012. In *Proc. TREC 2012 Microblog track*, 2012.

[40] A. Roegiest and G. V. Cormack. University of waterloo at TREC 2011: Microblog Track. In *Proc. TREC 2011 Microblog track*, 2011.

[41] G. P. Shapiro. *Discovery, analysis, and presentation of strong rules*, pages 229–238v. MIT Press, 2009.

[42] M. Sleeper, J. Cranshaw, P. G. Kelley, B. Ur, A. Acquisti, L. F. Cranor, and N. Sadeh. I read my Twitter the next morning and was astonished: A conversational perspective on Twitter regrets. In *CHI '13 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3277–3286, 2013.

[43] S. Sommer, A. Schieber, A. Hilbert, and K. Heinrich. Analyzing customer sentiments in microblogs: A topic-model-based approach for Twitter datasets. In *AMCIS 2011 Proceedings*, 2011.

[44] W. W. Stroup. *Generalized Linear Mixed Models: Modern Concepts, Methods and Application*. Chapman Hall/CRC Press, 2012.

[45] K. Tao, F. Abel, and C. Hauff. WISTUD at TREC 2011 Microblog Track: Exploiting background knowledge from dbpedia and news articles for search on twitter. In *Proc. of TREC'11*, 2011.

[46] C. J. van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33:106 – 119, 1977.

[47] B. Wang and X. Huang. Microblog TRACK 2011 of FDU. In *Proc. of TREC'11*, 2011.

[48] J. Wang and J. Zhu. On statistical analysis and optimization of information retrieval effectiveness metrics. In *Proc. of SIGIR'10*, 2010.

[49] L. Wenyin, X. Quan, M. Feng, and B. Qiu. A short modeling method combining semantic and statistical information. *Information Sciences*, 180:4031–4041, 2010.

[50] H. Wickham. *ggplot2: elegant graphics for data analysis*. Springer New York, 2009.

[51] Z. Ye, B. He, X. Huang, , and H. Lin. Revisiting rocchio's relevance feedback algorithm for probabilistic models. In *6th Asia Information Retrieval Societies Conference*, 2010.

[52] Z. Ye, J. Huang, and J. Miao. A hybrid model for ad-hoc information retrieval. In *SIGIR '12 Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval Pages*, 2012.

[53] C. Zhai. Statistical language models for information retrieval a critical review. *Foundations and Trends in Information Retrieval*, 2:137–213, 2008.

[54] Z. Zheng, H. Zha, K. Chen, and G. Sun. A regression framework for learning ranking functions using relative relevance judgments. In *Proc. of SIGIR'07*, 2007.

[55] F. Zhou, Z. Fan, Y. Bingru, and Y. Xingang. Research on short text classification algorithm based on statistics and rules. In *Third International Symposium on Electronic Commerce and Security*, 2010.

[56] B. Zhu, J. Gao, X. Han, C. Shi, S. Liu, Y. Liu, and X. Cheng. ICTNET at Microblog Track TREC 2012. In *Proc. TREC 2012 Microblog track*, 2012.