QATAR UNIVERSITY

College of Engineering

Department of Computer Science and Engineering

Hashtag-based Query Expansion for Ad-hoc Search in Twitter

Latifa Jaber F A Al-Marri

This thesis is submitted to Qatar University in partial
fulfillment of the requirements for the degree of
Master of Science in Computing

September 2015

Declaration

To the best of my knowledge, this thesis contains no material previously published or written by another person or institution, except where due reference is made in the text of the thesis. This thesis contains no material which has been accepted for the award of any other degree in any university or other institution.

Name Latifa Jaber F A Al-Marri

Signature ______________________________ Date __________

Committee

The thesis of **Latifa Jaber F A Al-Marri** was reviewed and approved by the following:

We, the committee members listed below, accept and approve the Thesis/Dissertation of the student named above. To the best of this committees knowledge, the Thesis/Dissertation conforms the requirements of Qatar University, and we endorse this Thesis for examination.

Memebers:

Dr. Ali Jaoua

Dr. Qutaibah Malluhi

Dr. Lynda Lechani

Supervisor:

Dr. Tamer Elsayed

Signature ______________________________ Date __________

Abstract

Twitter has increasingly become an important resource of information sharing. It is a platform that has multiple characteristics that differentiate it from other platforms and make it challenging for information retrieval. Given that tweets are short and user queries issued to retrieve relevant tweets are also generally short, this work tries to explore the effect of expanding queries using hashtags, since hashtags can generally give an idea of a tweet context. This has the potential to improve retrieval effectiveness of relevant tweets that would meet the user information need.

In this thesis, we propose two main ranking functions to retrieve relevant hashtags to a query to be used for expansion: KL divergence and Cosine similarity. We further study the use of temporal aspects by incorporating a decay temporal factor into the basic ranking functions.

We conducted experiments on both **TREC2011** and **TREC2013** test collections using the standard measures, *Precision at 30* (P@30) and *Mean Average Precision* (MAP), to evaluate the performance of the ad-hoc retrieval tasks in Twitter.

The main ranking functions are evaluated by measuring the recommended hashtags relevance with regard to the user query. The results showed that Cosine similarity with temporal decay factor is the most effective ranking function among the ones we experimented with where temporal decay factor improves the Cosine function performance by approximately 17% on average in recommending the best firstly ranked hashtag.

The proposed expansion model can be used to utilize hashtags or non-hashtag terms for expansion. The expansion model using hashtags-only have not improved the retrieval results compared to the baseline system, whereas, non-hashtag expansion model improves it by 4%. However, experiments showed that combining both models gives the best results with an improvement of 5%. In addition to that, it also improves the retrieval performance in both TREC collections and using both evaluation measures where the percentage of improvement in precision ranges from 3% to 9% and in MAP it ranges from

10% to 13% . Such improvement indicates that hashtags, a Twitter common feature, can be utilized effectively to improve search results of ad-hoc retrieval.

Contents

List of Tables

List of Figures

Acknowledgements

First and foremost, thanks to God that my prayers were answered and I was able to complete my thesis. Thanks also to my family, my mother, my son, sisters and husband. Thank you all for your patience, for all the times you tried to create a quite atmosphere for me to work, and for all your words that encouraged me and made be believe that I can do it despite all challenges. And eventually, thank you for your understanding and baring all the times I couldn't share with you.

Most importantly, deep thanks to my Supervisor, Dr. Tamer Elsayed, for all his support, persistence and encouragement and for all the help, guidance and advice. He has been most patient and helpful and kept following up to the smallest details. I appreciate all his efforts and I'm honored to be supervised by him.

Deep thanks also, to my friends who never stopped helping throughout the thesis (and judgments) and never stopped caring.

Dedication

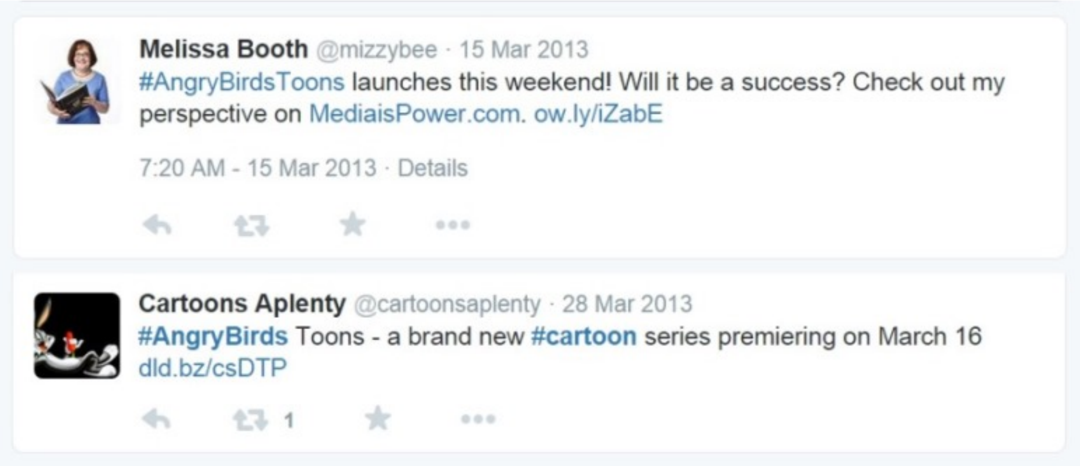
To my son Abdullah, who has been the most encouraging motive for me to accomplish my work. Even before you've seen the light, you gave me strength and power to do more. You give me the power, son, to climb mountains. I love you with all of my heart.

Chapter 1

Introduction

Microblogging services have become widely popular in recent years where users are able to share information, communicate with friends, and follow up on ongoing events. Twitter[1], in particular, is one of the most rapidly growing microblogging platforms. Twitter is a communication platform that has millions of worldwide users who post collectively over 400 million messages, called "tweets", per day [45].

Tweets are generally short and are limited to a maximum of 140-characters. Such limitation forces users to shorten tweets and post ones that may not be clear and might lack context. Twitter users have partially overcome this problem by using special symbols in their tweets such as '@' (to mention or reply to other users), 'RT' (to quote or re-post/re-tweet other tweets to followers), and '#' to tag the tweet by a following label, called a hashtag [20]. A hashtag is a word or phrase with no spaces prefixed with a hash symbol '#' inserted anywhere in the body of tweets [37] that generally indicates the topic of the tweet. Figure 1.1 shows examples of real tweets that contain relevant hashtags which explain the topic of the tweets. Note that the tweets are discussing the release of a new cartoon show called "Angry Birds" and the hashtags "#AngryBirdsToons", "#AngryBirds" and "#cartoon" added in the tweets give an understanding of the tweets context.



Melissa Booth @mizzybee · 15 Mar 2013
#AngryBirdsToons launches this weekend! Will it be a success? Check out my perspective on MediaisPower.com. ow.ly/iZabE

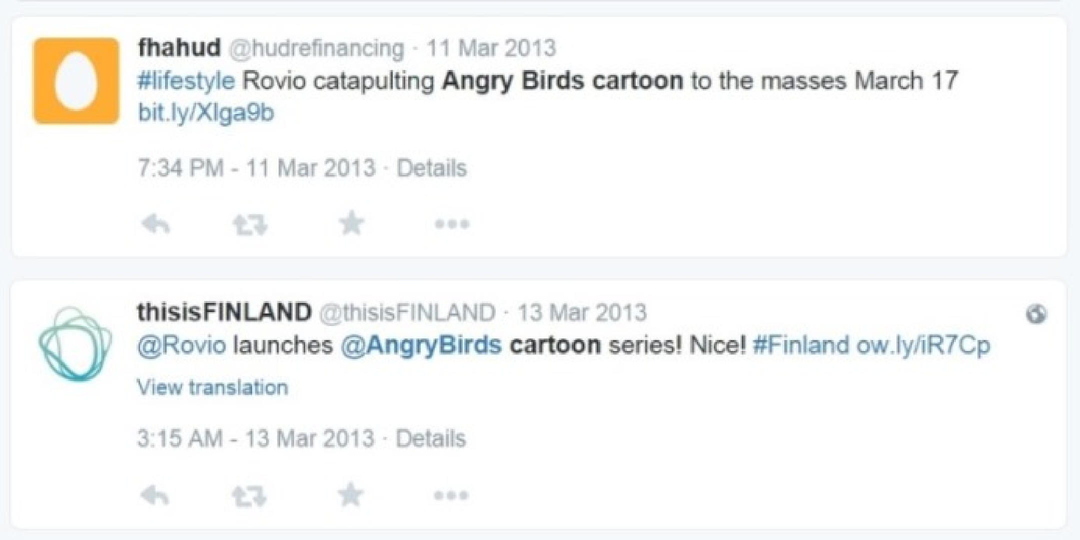
7:20 AM - 15 Mar 2013 · Details

Cartoons Aplenty @cartoonsaplenty · 28 Mar 2013
#AngryBirds Toons - a brand new #cartoon series premiering on March 16
dld.bz/csDTP

Figure 1.1: Sample tweets containing *relevant* hashtags.

[1] https://twitter.com/

Although hashtags are usually added as topic labels, they are sometimes misused. Figure 1.2 shows tweets discussing the same topic but with irrelevant hashtags that may be considered misleading. As shown in the tweets, the hashtags "#lifestyle" and "#Finland" are mentioned in the tweets and are not related to the cartoon show to be released.



fhahud @hudrefinancing · 11 Mar 2013
#lifestyle Rovio catapulting **Angry Birds cartoon** to the masses March 17 bit.ly/Xlga9b

7:34 PM - 11 Mar 2013 · Details

thisisFINLAND @thisisFINLAND · 13 Mar 2013
@Rovio launches **@AngryBirds cartoon** series! Nice! #Finland ow.ly/iR7Cp
View translation

3:15 AM - 13 Mar 2013 · Details

Figure 1.2: Sample tweets containing *irrelevant* hashtags.

1.1 Ad-hoc Search in Twitter

Twitter search is becoming more needed recently to get updated on real-time incidents or topics over tweets that are posted every moment. Users want to get up to speed on topics, and thus wish to retrieve highly relevant tweets that will provide context, and perhaps link to important external resources [42]. Typically the process of an ad-hoc search in Twitter starts with a user issuing a query reflecting his/her information need and the retrieval system would retrieve a list of ranked tweets ordered by their possible relevance to the query.

However, having an effective retrieval of recent and relevant tweets for an ad-hoc search process in Twitter poses many challenges. There are multiple characteristics that differentiates Twitter from any other platform and makes results retrieval for a user query a challenging problem. Some of these challenges are due to the nature of tweets that are very short and may involve a number of people. Also, tweets have temporal aspects i.e., tweets might discuss an event or a topic that is temporal and focused in a certain period of time. Another difficulty is the short length of queries issued by users (2 words on average [44]) and hence queries may not be informative of the user information need. All of these special characteristics and difficulties make ad-hoc results retrieval in Twitter an interesting but challenging research problem.

Improving the retrieval process of a user's issued query in Twitter has been a research problem that multiple studies have investigated. One of the main techniques that can be used to improve a retrieval system performance is called *query expansion*. Query expansion is a technique that can be used to enrich the representation of a user query by expanding it with terms that are considered relevant to the original query. Given the fact that user's queries in Twitter are generally short, such technique proved to be experimentally effective [3, 8, 18, 26, 35].

One of the most popular features of Twitter is *Hashtags*. Hashtags can give an idea of a tweet context, any topic in Twitter may propagate quickly among users through the use of hashtags which can create a community of users with similar interests. Upon that, multiple studies focused on hashtags through recommending, classifying and representing them (details of such studies can be found in the background and related work chapter 2). In addition to that, hashtags can also be utilized in query expansion.

In this work, we experiment with the use of hashtags for ad-hoc retrieval in Twitter. In that context, we tackle two main problems: recommending hashtags for a given query, and using hashtags for query expansion. The method proposed mainly builds a model for the user query and a model for all hashtags that are potentially relevant to the query. We experimented with multiple ranking functions by which hashtags can be selected which are further elaborated in chapter 3.

1.2 Research Questions

In this work, we try to explore the answers for the following research questions:

RQ1: Which ranking function is most effective for relevant hashtags retrieval?
Having a set of potentially-relevant hashtags for a given query, a ranking function is used to rank the set in order of relevance to the query. We study multiple ranking functions for that purpose.

RQ2: Can we recommend better hashtags if we consider temporal aspects?
Discussions in Twitter are naturally temporal. Upon such nature, we explore the temporal aspect by incorporating it in our ranking functions. We try to study recommending hashtags that are temporally closer to the query time and whether or not they would be more relevant.

RQ3: How can hashtags be represented?
In order to measure the hashtag relevance to a given query, a model is created to represent both the hashtag and the query. We try to investigate the representation models, especially the hashtag model, and how they affect the recommendations.

RQ4: How effective is query expansion with hashtags?
Given that several studies explored the use of query expansion in microblog search [3, 8, 18, 26, 35], we try to use the query expansion model but with the use of hashtags for expansion. The model used can incorporate both hashtags and non-hashtag terms. The difference between the use of non-hashtag expansion terms or hashtags, is a key question that is experimented in this work.

RQ5: How does the expansion model preform over different test collections?
It is imperative to measure the performance of the expansion model on different test collections that have different tweets and total size to have a better insight on the model performance given different test conditions.

1.3 Contributions

The contributions of this work are summarized as follows:

1. To the best of our knowledge, there are few studies exploring query expansion using hashtags [12]. This work gives an elaborated study of using hashtags to expand any user query in the context of microblog search.

2. We experimented with different ranking functions which serve in hashtag recommendation with regard to a user query. In addition to that, this work provides functions that incorporate temporal aspect to rank hashtags.

3. Each hashtag ranking function was evaluated by labeling hashtags relevance with regard to the user query. These judgments can be shared to provide a good experimental evaluation basis for any future studies or experiments in microblog search.

4. Unlike experiments of other studies that focus on the same problem [12], the experiments of this work were conducted on a large collection of tweets. We used two of the most-widely known microblog test collections (**TREC2011** [38, 43] and **TREC2013** [28, 29]). Those collections contain millions of tweets and about 223 experimental queries that make them well suited for experiments.

1.4 Thesis Organization

This thesis is divided into five chapters including the introduction chapter. The next chapters are organized as follows:

- **CHAPTER 2** provides a background information on query expansion and literature review of studies that focus on hashtags.

- **CHAPTER 3** explains in details the methodology we propose.

- **CHAPTER 4** illustrates the experiments conducted to answer the research questions and presents an analysis of the results.

- **CHAPTER 5** gives a conclusion of our work and some future work suggestions that can be further explored.

Chapter 2

Background and Related Work

This chapter highlights the different aspects in which hashtags were used in microblog search and provides a discussion of existing studies in literature that focus on the use of hashtags for different problems in the microblog domain.
Section 2.1 gives a background information on query expansion and the different methods that some studies proposed to expand a query. In section 2.2 hashtag retrieval for query expansion is explained whereas section 2.3.1 presents hashtag recommendation and how different studies propose recommending hashtags. Section 2.3.2 discusses the different ways to represent hashtags and section 2.3.3 introduces hashtag classification.

2.1 Query Expansion

Usually queries submitted to Twitter are short. This poses a severe vocabulary mismatch problem rooted in the sparsity of tweets. For this reason, expansion techniques can be applied to enrich the representation of a query. Pseudo relevance feedback (PRF) is a well-known method where a query is submitted to retrieve initial results set then such set is used to extract expansion terms to be added to the original query. Query expansion has been explored in research studies to improve retrieval results in many different domains such as the web [7, 41, 46], multimedia [16, 23, 36], news [21] and others. In addition to that, using PRF for query expansion has shown good retrieval effectiveness in microblog search [3, 8, 18, 26, 35].

Some researchers tried improving PRF performance with different techniques. For example, Whiting et al. [48] proposed an improved PRF query expansion by the selection of n-grams extracted from feedback tweets based on the strength of their temporal correlation with other extracted n-grams (i.e. temporal evidence), in combination with their Term Frequency (TF) within the feedback tweet set (i.e., TF evidence). Whiting et al. model evaluation showed that neither temporal nor TF evidence alone is consistently able to perform optimally, whereas combining both evidence sources leads to, on average, better retrieval performance.

Given that microblog services focus on recent issues and since users in a microblog community can express opinions and discuss social issues with other users immediately, incorporating time information into ranking is crucial in microblog retrieval. Choi et al. [8] proposed a time-based relevance model which incorporates time factor into pseudo relevance model framework. The proposed model allows extracting the expanded terms from the relevant tweets published in a time period of the query. Such time period is suggested by Choi et al. by estimating the time period when an event (query topic) happened and people discussed that issue heavily in the past. This time period is considered as a relevant time period for the query. In other words, PRF was constructed with the tweets that occurred in the most active days in terms of retweeting. Choi et al. approach for query expansion improved retrieval performance [8].

Some methods tried to expand queries with information from external resources other than tweets. Louvan et al. [31] used PRF model where the main scoring function for expansion terms used was TF(Term Frequency)-IDF(Inverse Document Frequency) . But they also tried to expand the original query, not with top-ranked retrieved tweets, but with snippets from Google search results. Evaluation results showed that query expansion using an external resource like Google snippets yields to the best experimental results. Such outcome was explained by Louvan et al. that Google snippets can give richer terms than internal dataset (which usually contain nonstandard terms and abbreviations), so it can help get higher relevant tweets. Similarly El-Ganainy, Tarek et al. [15] used PRF for query expansion but also incorporated search results from Google in the query expansion terms. The final expanded query contained the following: the terms of the top retrieved tweets from the results of submitting the original query. Also, the title of the top search web page. In addition to that, the top terms from top retrieved Google search results are also added to the expanded query where the optimal number was found to be 3 web pages.

Unlike the previously described methods of query expansion, Le et al. [25] performed a reformulation of the original query by using a learning model where an SVM[1] classifier is used. After the initial retrieval of search results from the original query, the classifier will label the tweets into "relevant" or "non-relevant" then the classifier prediction learning process goes on until the performance reaches a reasonably good point. After that the system will prepare a new query based on the learned model and this new query will retrieve the set of tweets. Such system proved to be the best performing system in TREC[2] 2013.

PRF assumes that the top ranked documents in the initial search results are relevant and that they contain topic-related words appropriate for relevance feedback. However, those assumptions do not always hold in reality because the initial search results often contain many irrelevant documents. Hence, some researchers tried a two-stage PRF query expansion [27, 35, 40]. Liang et al. [27] retrieves the top 10 tweets in the first PRF stage and from those tweets one tweet would be selected to be submitted as a query for the second stage of PRF query expansion. There are three types of tweets selection that were experimented: (**LT**) the tweet with the latest timestamp. (**ET**) the tweet with the earliest timestamp. (**HS**) the tweet with the highest score of similarity to the query. The evaluation of those three first stage tweet selections showed that choosing the tweets with the highest similarity score (**HS**) gives the best results. Furthermore, evaluation results

[1] SVM (support vector machine) is a learning model which analyzes data and recognizes patterns. SVM is usually used for classification and regression analysis.

[2] http://trec.nist.gov/. TREC is a core Information Retrieval (IR) evaluation conference that organizes research competitions on several tracks including microblog track that focuses on search-related tasks in Twitter.

also showed that two-stage PRF query expansion significantly improve the performance in comparison to first-stage PRF query expansion and to the basic retrieval system.
Miyanishi et al. [35] also proposed two-stage relevance feedback approach for microblog search using tweet selection feedback in which the user selects only one relevant tweet among top ranked initial search results and its combined with the original user query for tweet selection feedback. The final evaluation and experiments showed that tweet selection feedback improves the retrieval performance.
Based on Liang et al. [27] work, Qiang et al. [40] implemented the Real-time Tweet Ranking (RTR) model which incorporated a two-stage PRF query expansion technique in addition to document expansion using shortened URLs to enrich document representation and temporal re-ranking functions for better retrieval results.

2.2 Hashtag Retrieval for Query Expansion

Up to our knowledge, not many studies were performed with regard of expanding a query using hashtags. Efron [12] is the main research study that explored expanding queries with hashtags. Efron proposed an approach for retrieving hashtags on a topic of interest to a searcher. The task involves accepting a query and returning a ranked list of hashtags using Kullback–Leibler (KL) divergence [22] as a ranking function. Efron experimented with PRF query expansion by retrieving the top hashtags and adding them to the query using different weighting schemes. Efron evaluated the work on a collected data using Twitter's streaming API[3]. The data was collected on a 24-hours interval having over 3 million tweets and 39 query topics. The evaluation based on the data collected showed that using hashtag-based feedback gives results that are statistically and significantly better than the baseline run which uses standard term-based feedback.

Anagnostopoulos and Mylonas [4] proposed expanding a query with related twitter hashtags by creating a social semantic network derived from the user query terms. Each query term is used to retrieve tweets that are related in a specific period in which hashtags are extracted from. Next to create the semantic network, a specific semantic weight is calculated between each hashtag and query term. Eventually, top weighted hashtags can be used to expand the query and the same process can be iterated using the new expanded query. Anagnostopoulos and Mylonas [4] believed that such social semantic network can be dynamically created to be capable of suggesting related terms to users during their web search. Based on that, they tested their methodology on a case study and compared their expanded query to top search engines where the method used showed promising results.

2.3 Hashtags in Other Tasks

2.3.1 Hashtag Recommendation

Many studies focusing on hashtag recommendation were conducted and many methods were used. These studies and methods are further discussed in this section.

[3] http://api.twitter.com

Weng et al [47] recommends hashtags for a given stream of user's posts (tweets). The method used extracts candidate hashtags from a discussion graph. Weng et al. represents the discussion graph using the social connections and the hashtag interactions among users. The social connections are represented with a direct graph where V represents a set of users and edges are defined as $E^{sc} = \{(v_a, v_b)|v_a, v_b \in V \text{and}\, v_a\, \text{follows}\, v_b\}$. The edge weight is calculated over the period of time between the posts of users v_a and v_b using the same hashtag. Such weight w_{ab} captures the influence that user v_a receives from v_b in the use of the same hashtag. Eventually, an interestingness-measuring function is defined using a learning to rank approach in which hashtags with high interest are recommended.

Bansal et al. [5] retrieve a ranked list of hashtags given a user query by exploring semantic information of hashtags. Bansal et al. [5] propose "Semantically Enriched Microblog Document (SEMD)" structure, which enables semantic retrieval of hashtags. SEMD presents a machine learning based approach to segment the hashtags and link them to Wikipedia. This allows to extract latent semantic information about hashtags. For a given user query, a list of top 500 tweets is obtained and ranked by their relevance according to SEMD structure. In order to recommend most relevant hashtags, multiple hashtag ranking approaches were proposed: *GlobalRank (GR)*, where the hashtags in top 500 retrieved tweets were ranked based on their frequency in the overall collection. *RetrievedHashtagRank (RHR)*, where the hashtags in top 500 retrieved tweets were ranked based on their frequency in top 500 tweets. TF-IDF and *KLDivergenceRank (KLDR)*, where KL Divergence was used. Bansal et al. [5] observed that KL-divergence performed significantly better than the other proposed approaches.

Cosine Similarity is a method that is used in some studies to recommend hashtags. Xiao et al. [49] and Kywe et al. [24] both used cosine similarity but with different approaches. Xiao et al. created a hashtag vector based on tweets that contain the same hashtag and calculated the similarity between the hashtag vector and a news topic vector where hashtags with high similarity scores are recommended for the news topic. Kywe et al. [24] on the other hand, used the top most similar users and tweets to a given user tweet using Cosine similarity to propose a personalized hashtags recommendation.

Both Zangerle et al. [50] and Harvey et al. [19] proposed recommending hashtags to any entered user tweet. Zangerle et al. [50] retrieve a set of similar tweets using TF-IDF and eventually extract candidate hashtags from them. Harvey et al. [19] did the same but used cosine similarity as a scoring function instead. For hashtag ranking Zangerle et al. [50] experimented multiple methods. *OverallPopularityRank* where hashtags with the highest number of occurrences in the collection are ranked higher. *RecommendationPopularityRank* hashtags are ranked according to the number of occurrences in the similar tweets set and *SimilarityRank* where hashtags belonging to tweets with the highest similarity score are ranked higher. Zangerle et al. [50] found that *SimilarityRank* proved to be the best performing ranking function. Whereas Harvey et al. [19] further improved the same ranking function by including temporal information in rankings.

Given the fact that hashtags are usually used to describe a tweet's topic, some studies concentrated on the use of LDA (Latent Dirichlet Allocation)[4] to recommend hashtags.

[4] LDA (Latent Dirichlet allocation) is a generative probabilistic model in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled

Godin and Slavkovikj [17] and Ding et al. [11] for instance, used LDA in addition to Gibbs sampling[5] algorithm to determine a tweet's topic. Once a tweet's topic is determined, the LDA topic-term distribution can be used to extract the top words from the topic. Hence, the extracted words can resemble the general topic of the tweet and is helpful for recommendation.

2.3.2 Hashtag Representation

As mentioned before in the introduction chapter 1, when hashtags are created they must not contain spaces between words, and this may decrease readability. Some hashtags, moreover, use abbreviations and others are created with slang language. Given such nature of hashtags and how they can be created, users may find it difficult to understand the meaning of them. Thus, some research studies focused on hashtag representation in which researchers tried to give an effective method to represent hashtags in a way that would make it more easier for users to comprehend.

Qiang et al. [40] changed the representation of all hashtags as part of their tweet's collection preprocessing. Hashtags were segmented, that is all connected words in a hashtag are further broken apart and spaces were added between them. Segmentation was done with the use of an English dictionary generated from the same collection in addition to the use of a MaxMatch algorithm [40].

LDA can also be utilized for hashtag representation. Ma et al. [32] extended LDA and proposed Tag-Latent Dirichlet Allocation (TLDA) which can be considered a new approach to bridge hashtags and topics. TLDA incorporates hashtags in the generative process particularly when generating a tweet, a subset of hashtags are selected based on the mutinomial distribution of hashtags in this tweet. Then, under the chosen hashtag, a topic is sampled from a multinomial distribution which is assumed to be generated from a symmetric Dirichlet distribution. Topics in TLDA are also described as a multinomial distributions over the vocabulary [32]. In general, TLDA model can populate the hidden topic structures for each hashtag.

2.3.3 Hashtag Classification

Hashtag classification becomes a necessity when a hashtag is describing two different events or is basically ambiguous, in other words, if a hashtag is referring to different objects. For example, the hashtag #apple may refer to the fruit or either to the technology company.

There are a couple of attributes, characteristics, that can be associated to Hashtags and can be utilized to classify them. Cui et al. [10] gathered the following attributes to develop a categorization algorithm:

1. Hashtag instability which is calculated using Gaussian distribution.

as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document [6].

[5] Gibbs sampling is a technique used to rapidly explore the space around a target distribution using repeated sampling [17].

2. Twitter Meme Possibility (TMP) which is measured using the length and the position of the hashtag.

3. Authorship Entropy that is measured by the number of hashtag contributors (authors).

Such method was experimentally proven to be effective in categorizing hashtags [10] and was used for removing the advertising hashtags, which may cause false alarms on events.

Another aspect in which hashtag classification comes useful is to predict hashtag popularity to identify fast emerging topics attracting collective attention. To classify a hashtag to be popular, content and context features can be studied to define popularity. Ma et al. [33] composed something called a hashtag profile. A hashtag profile T_t^h is the collection of tweets annotated by hashtag h in time interval t. To represent the content features of a hashtag, the hashtag profile is used to generate a 20-dimension topic vector using LDA that represents the topic distribution of the hashtag. The entries in the vector would be quantifying the likelihood of the hashtag belonging to a corresponding topic. For context features, a community graph is created using T_t^h. There are 8 context features derived from the graph to capture the current popularity of a hashtag, the influential power of its users, the connectivity among these users, and the distribution of the users exposed to it [33]. Both content and context features were effective in Ma et al.'s experiments [33] and it is worth mentioning that context features were relatively more effective than content features.

2.4 Summary

In this chapter, a background information is provided explaining query expansion and the different research studies that tried to improve it using multiple schemes. In addition to that, a literature review was presented, focusing on the different conditions in which hashtags are used. Mainly in query expansion to improve retrieval, hashtag recommendation, representation, and classification. There were different studies and methods that were discussed in this chapter with regard to each aspect in which hashtags are the main focus. The next chapter will describe the approach of this work in details.

Chapter 3

Hashtag-based Query Expansion

A thorough description of the problem that this work is focusing on is clarified in this chapter in addition to the methodology and its different steps and variants.

3.1 Overview

Twitter has increasingly become an important resource of information seeking. It is a platform that has some characteristics that differentiate it from other platforms and makes it a more challenging platform for information retrieval. Given that tweets are short and user queries are also generally short, this work tries to explore the effect of expanding queries. As mentioned in the query expansion background section 2.1, Pseudo Relevance Feedback (PRF) is a well-known method for query expansion and it experimentally proven to be effective in microblog search [3, 8, 18, 26, 35]. This work mainly focuses on expanding queries using one of the most common twitter features which is *hashtags*. Hashtags can usually describe the content or the topic of a tweet and it was shown that hashtags can improve retrieval results if used effectively with query expansion [12]. Generally, our work is based on Efron [12] work described earlier in the background and related work chapter (section 2.2).

The goal of our work is to explore ways of utilizing hashtags for query expansion. When a user submits a query, we would like to find a list of hashtags that are relevant to the query information need. The following sections will give details of the steps to achieve our goal and the steps are:

1. **Building hashtag model:** section 3.2 gives an explanation on how the hashtag model is created using vector representation. In addition to that, the vector creation and vector terms weight's calculation is also explained in the same section.

2. **Building query model:** similar to the hashtag model, a query model is created using vector representation. The details of how the query model is represented is further described in section 3.3.

3. **Approximating query and hashtag models**: section 3.4 gives an explanation on why approximation is used in the models created and a clarification on how they are approximated.

4. **Ranking top hashtags:** in section 3.5 several ranking methods are explained to rank top hashtags with regard to a query using the previously created hashtag and query models.

5. **Weighting top hashtags for query expansion:** hashtags are eventually weighted for query expansion using different weighting schemes which are explained in section 3.6.

6. **Weighting top expansion terms for query expansion:** expansion terms are also used in our proposed feedback model and section 3.7 gives a description of how expansion terms are retrieved and weighted.

7. **Building feedback query model:** the detailed final model used for query expansion and its variants is explained in section 3.8.

3.2 Hashtag Model

First of all, a language model is defined for each hashtag h_i in the collection. To be specific, for each hashtag h_i a vector Θ_h is created that represents a multinomial distribution over the vocabulary of words in the collection. Each word (w) is represented with a probability score $P_{ml}(w|h_i)$ which can be called (θ_{wh}). Eventually the vector model looks something like this: $\Theta_h =< \theta_{1h}, \theta_{2h}, ..., \theta_{nh} >$.
(θ_{wh}), the term score in Θ_h, is calculated using the *Maximum Likelihood Estimator* (**MLE**) [9, 34] as the following:

$$P_{ml}(w|h_i) = \theta_{wh} = \frac{tf}{|T_{h_i}|} \tag{3.1}$$

Where $|T_{h_i}|$ is the sum of length, the total word count, of the top tweets that contain the hashtag h_i. T_{h_i} is retrieved by submitting the hashtag h_i as a query in which the top tweets would express T_{h_i}. The number of top tweets is defined with a variant (**MLE_h_tweets**) that can be experimentally changed to any value. tf, on the other hand, is the term's frequency within T_{h_i}. In other words, the number of occurrences of the word w in the set of tweets containing the hashtag h_i.

But in equation 3.1 above, words that do not appear in the set T_{h_i} will have a value of zero as if they are ignored in any further calculations. Such words may have importance in further calculation with regard to the query for instance. Hence, there is a need to give a value, even if it is a small one, other than zero. For that, a method that's called *smoothing* can be used. Therefore, we use smoothing in the calculation of weights in the hashtag vector Θ_h. There are several smoothing methods but in this work we smooth the estimated model by Bayesian updating with Dirichlet priors [34, 51] where the parameter μ is set to 2000 as in Efron [12] research. Accordingly, the equation above 3.1 will be changed to use smoothing to become:

$$\theta_{wh} = \frac{|T_{h_i}|}{|T_{h_i}| + \mu} P_{ml}(w|h_i) + \frac{\mu}{|T_{h_i}| + \mu} P(w|C) \tag{3.2}$$

Where $P(w|C)$ is the probability of the word w given the whole collection C. To be specific, it represents the count of the word w in the collection, collection frequency cf [9, 34], divided by the collection length $|C|$:

$$P(w|C) = \frac{cf}{|C|} \tag{3.3}$$

It is worth mentioning that any retrieval task mentioned through out this thesis, such as submitting hashtag h_i to retrieve T_{h_i} uses *query likelihood model* [39]. The query likelihood model, $P(D|Q)$, ranks documents by the probability that the query text could be generated by the document language model [9]. Using Bayes' rule it is calculated as the following:

$$P(D|Q) \propto P(Q|D)P(D) \tag{3.4}$$

$P(D)$ is considered uniform and would not affect the ranking, therefore it can be ignored from the equation above 3.4. $P(Q|D)$ can be calculated as the following [9]:

$$P(Q|D) = \prod_{i=1}^{n} P(q_i|D) \tag{3.5}$$

where $P(q_i|D)$ can be computed using MLE.

3.3 Query Model

Similar to the hashtag model, the query is represented with a vector Θ_q where each word w is represented in the vector with a weight using MLE and with the same smoothing method. The weight of a word in the query vector θ_{wq} is calculated as shown in the following equation:

$$\theta_{wq} = \frac{|T_q|}{|T_q| + \mu} P_{ml}(w|q) + \frac{\mu}{|q| + \mu} P(w|C) \tag{3.6}$$

Where $|T_q|$ is the total sum of the length of words in top tweets retrieved by submitting the query q. Number of tweets in $|T_q|$ is defined by a variant the can be assigned to any positive number, named (**MLE_q_tweets**). $P_{ml}(w|q)$ is the MLE of the word w in the query q calculated as: $\frac{tf_w}{|T_q|}$. Where the term frequency tf_w of w is calculated within the tweets retrieved from submitting the query q.

3.4 Approximating Query and Hashtag Models

The hashtag representation model is presumably created for all hashtags in the collection, as in Efron [12]. However, that can be time consuming. Keeping in mind that each hashtag should be submitted as a query to calculate each vector terms's weight and this includes a lot of query results retrieval and can be inefficient. Hence, approximation can be used to have a more efficient model representation. Approximation is used in two aspects: obtaining hashtags and the vocabulary of words in the collection. Figure 3.1

shows how to get the set of hashtags and words to be able to create the representation model.

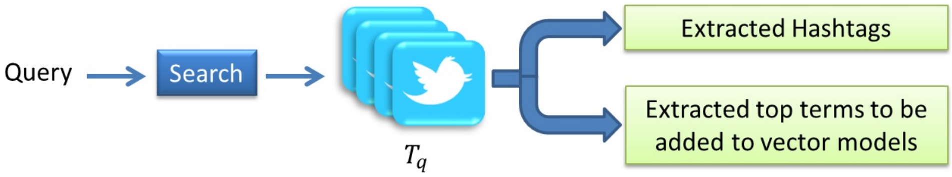


Figure 3.1: Abstraction on how approximation is done by extracting hashtags and terms from the original query results set

3.4.1 Hashtags Extraction

Using the original query results (T_q) we can extract a list of all hashtags that appeared in the result set to be our main hashtags list to score and rank. The number of results to extract hashtags from is controlled by a free parameter, called (**h_extr**).

3.4.2 Terms Extraction

Like hashtags extraction and from the same results set we can also extract the top terms (words) using TF-IDF to be our main vector terms in addition the the query terms and that would be an approximation representing the whole vocabulary of words. TF-IDF is computed as the following:

$$TF - IDF = (tf_w)(idf(w)) = (tf_w)\left(\log \frac{N}{df_w}\right) \tag{3.7}$$

tf_w represents the number of occurrences of the word w in the set of tweets retrieved by the original query. $idf(w)$ is the *inverse document frequency* of the word w [9, 34] which indicates that the word is rare if the value is high and the opposite is true. $idf(w)$ is computed by $\log \frac{N}{df_w}$ where N is the total number of documents in the collection and df_w, *document frequency* of the word w, indicates the number of documents that contain the word w [9, 34].

Those terms that are extracted and eventually added to the vector models are controlled by a threshold (**vector_nTerms**). For example, if the threshold is set to be 20 and the original query has 5 terms, then the total number of terms in the vector representation of both the query and the hashtag model would be 25.

3.5 Ranking Top Hashtags

Up to this point, a model of each hashtag h_i, called Θ_h, and for the query Θ_q is created. Now, we can measure the similarity between each of the hashtag model and the query model and get the ranking of the top similar and most relevant hashtags. In this work, we try three ranking methods:

- Kullback–Leibler (**KL**) divergence.

- Cosine Similarity.
- Temporal Decay

KL divergence is the main ranking function used in Efron's [12] work. In this work we try to introduce and explore other ranking functions which are Cosine Similarity and Temporal Decay.

3.5.1 Kullback–Leibler (KL) divergence

Kullback–Leibler (**KL**) divergence [22] can be used as a ranking equation. KL is usually used to measure the divergence between two probability distributions. Here we need KL to measure the similarity rather than the divergence between both models Θ_h and Θ_q. Hence, the *negative* **KL** is used to rank hashtags in a decreasing order. The KL ranking equation is computed as follows:

$$r(h_i, q) = -KL(\Theta_q \| \Theta_h) = \sum_{j=1}^{n} P(q_j) \ln \frac{P(q_j)}{P(h_j)} \tag{3.8}$$

In the equation above 3.8, the summation starts from 1 until n; where n resembles the total number of terms in each of the hashtag and query vector models. $P(q_j)$ is the probability of the query model q for the term j and $P(h_j)$ is the probability of the hashtag model for the same term j. In other words, both probabilities can be substituted by θ_{wq} and θ_{wh} respectively (explained earlier in sections 3.2 and 3.3) and the equation would become:

$$r(h_i, q) = -KL(\Theta_q \| \Theta_h) = \sum_{j=1}^{n} \theta_{jq} \ln \frac{\theta_{jq}}{\theta_{jh}} \tag{3.9}$$

3.5.2 Cosine Similarity (Cosine)

Cosine similarity is a known similarity measure where the similarity between two vectors is measured by calculating the cosine of the angle between them [9, 34]. Cosine similarity can be used to rank similar hashtags to a given query as the following:

$$r(h_i, q) = sim(\Theta_q, \Theta_h) = \frac{\Theta_q . \Theta_h}{\|\Theta_q\| \|\Theta_h\|} = \frac{\sum_{j=1}^{n} (\theta_{jq})(\theta_{jh})}{\left(\sqrt{\sum_{j=1}^{n} (\theta_{jq})^2}\right) \left(\sqrt{\sum_{j=1}^{n} (\theta_{jh})^2}\right)} \tag{3.10}$$

Where the numerator represents the dot product, also known as the inner product, of the vectors Θ_q and Θ_h and the denominator is the product of their Euclidean lengths.

3.5.3 Temporal Decay

Temporal retrieval of results in microblog ad-hoc search domain is a very important aspect in which the recent the tweet is to the given query the more relevant it would be [8]. The reason is the nature of the microblog domain and how it may reflect users events discussions and interactions. Building on such observation, we explore the temporal aspect by incorporating it in our ranking methods.

In this work, we use an exponential temporal decay function to re-rank the list of hashtags obtained from the ranking functions, $r(h_i, q)$, explained in section 3.5 above. If we denote the new temporal ranking function as $r_{temporal}(h_i, q)$ then, and as in [13, 14, 20], the temporal decay function can be expressed as follows:

$$r_{temporal}(h_i, q) = s_{h_i}.r.e^{-r.|t_{h_i}-t|} \tag{3.11}$$

Where r is an exponential rate parameter that controls the temporal influence which is set to 0.01. s_{h_i} is the ranking score of the hashtag calculated using one of the ranking functions explained in section 3.5 either **KL** divergence to be named **KL** divergence with Temporal decay (**KL_t**) or Cosine similarity to be named Cosine similarity with temporal decay (**Cosine_t**). Whereas, $|t_{h_i} - t|$ denotes the time difference between the hashtag time t_{h_i} and the query time t. The hashtag time t_{h_i} can be represented in different ways and in this work we try two types of hashtag time representation:

1. The hashtag time can be equal to the timestamp of the most recent tweet (with regard to the query time) that the hashtag was extracted from in the approximation methodology.

2. The hashtag time can be equal to the average timestamps of tweets timings in which the hashtag got extracted from.

The following Figure 3.2 gives an explanation of the two representations.

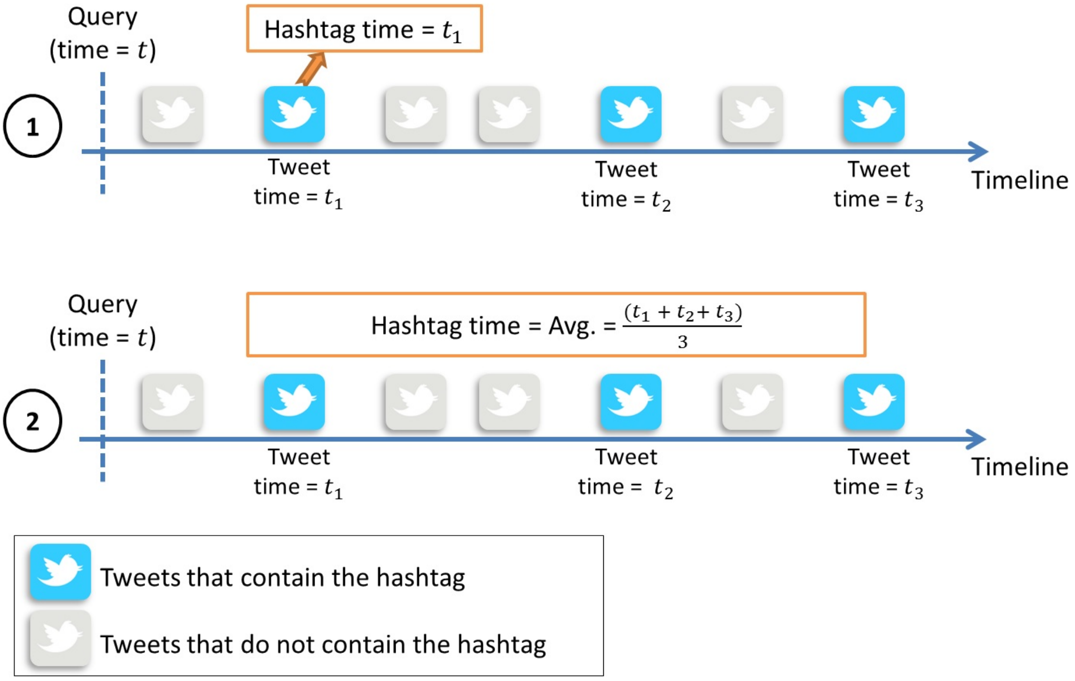


Figure 3.2: The two different ways of representing hashtag time. In (1), the hashtag time is equal to the timestamp of the most recent tweet that contains the hashtag. In (2), the hashtag time is equal to the average timestamps of all tweets that contains the hashtag.

3.6 Weighting Top Hashtags

Given the ranked hashtags list, we can select k top ranked hashtags to expand the query q. There are three methods of weighting the expansion hashtags:

- **HFB1**: using a uniformly distributed weights.
- **HFB2**: each term is proportional to $\frac{IDF(h_i)}{maxIDF}$; where IDF is the inverse document frequency [34] which is calculated as the following:

  $$idf(h_i) = \log \frac{N}{df_{h_i}} \tag{3.12}$$

  where df_{h_i} is the document frequency of the hashtag h_i in the whole collection and N is the total number of documents (tweets) in the collection. $maxIDF$ is the IDF for a hashtag with document frequency 1 which would be: $maxIDF = \log N$.
- **Ranking_score**: both HFB1 and HFB2 are the main weighting schemes experimented in Efron [12]. We also experiment with the actual KL or Cosine ranking score which can be used to weight hashtags in the expanded query.

3.7 Expansion Terms

To experiment more with query expansion, we have considered expansion with other terms than hashtags. The top terms from the initial results of the original query (T_q) are extracted using $TF - IDF = tf.idf$ [34]. But tf here calculates the number of occurrences of a term within the query results T_q because those results T_q are assumed to be relevant to the query and hence expansion terms are extracted from them. Such scoring is used to favor expanding a query using terms that frequently appeared in top retrieved tweets. The terms are thus viewed to be the most relevant to the original query [20].

Given that, the terms are used to expand the query having the same TF-IDF weight but with normalization. Normalization is calculated by diving on the sum of the expansion term's weights. We call the model containing top expansion terms with normalized weights Θ_t.

3.8 Feedback Query Model

Efron [12] focused on hashtags only in the feedback model he presented. In our work and now that we have the top hashtags and terms to expand with, we combine both of the expansion methods in one feedback model similarly to the work conducted by Liu and Croft [30]. The model is represented as the following:

$$\Theta_{fb} = (1 - \alpha - \beta)\Theta_q + \alpha\Theta_t + \beta\Theta_h \tag{3.13}$$

Where both α and β are tunable parameters in which values ranges from 0 to 1. Values selected should eventually sum up to 1 for all tuning parameters: $(1 - \alpha - \beta)$, in addition to α and β. If α is given a higher value than β then, this means that the expansion terms would have a higher importance and effect over the hashtags and the opposite setting stands.

Each part in the equation above 3.13 represents a component of the expanded query. To be specific, Θ_q represents the original query terms in which weights are eventually set to be uniformly distributed. Θ_t expresses the top expansion terms weighted with the normalized TF-IDF score as mentioned before in section 3.7. And Θ_h resembles the top expansion hashtags with weights set to use either HFB1, HFB2, or Ranking_score explained in section 3.6. Figure 3.3 gives an illustration of the main three components of the feedback model.

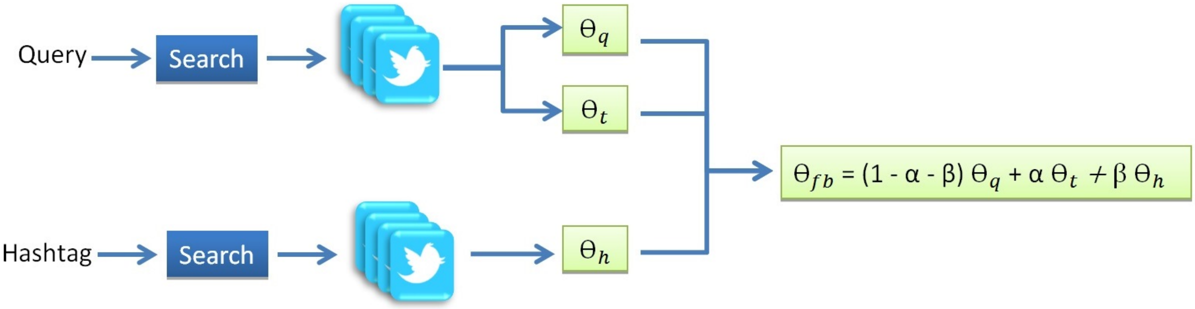

Query
Search
Θ_q
Θ_t
Θ_{fb} = (1 - α - β) Θ_q + α Θ_t + β Θ_h
Hashtag
Search
Θ_h

Figure 3.3: Illustration of the three main components of the query feedback model

If one of the model components is equal to the other then the sum of the weights as the equation denotes will be calculated. For instance, if an expansion hashtag is equal to a query term, then the weight of the query term Θ_q will be substituted in the feedback equation 3.13 in addition to the hashtag weigh Θ_h and eventually both components $(1 - \alpha - \beta)\Theta_q$ and $\beta\Theta_h$ are summed together to produce the new weight of the term. The third component, expansion terms Θ_t, is ignored and would be equal to zero.

Chapter 4

Experimental Evaluation

In this chapter, we present and analyze the results of our conducted experiments. In section 4.1, we discuss the evaluation setup detailing the datasets (collections), retrieval model, and the evaluation measures used to evaluate the results. Furthermore, in section 4.2, we present the evaluation results answering a set of research questions that focus on evaluating the implemented methodology. Finally in section 4.3, a comparison between this study and Efron's [12], the main research study this work is based on, is presented to detail the main different outcomes of the evaluation.

4.1 Setup

This section gives details on the datasets used to evaluate the proposed model. In addition to that, an explanation of both the retrieval model used to retrieve results and the evaluation measures used to evaluate the results are presented.

4.1.1 Datasets

There are a couple of datasets (collections) provided by TREC for the microblog track which are widely known and used in research. For our experiments, we used both **TREC2011** [38, 43] and **TREC2013** [28, 29] collections used in microblog track[1]. Such collections give a large number of tweets, 16 million in **TREC2011** [38] and 243 million tweets in **TREC2013**, that make them well suited for experiments. In addition to that, the track provided multiple queries (topics) along with their relevance judgments that are needed for evaluation. The **TREC2011** collection has two query sets available (2011 and 2012) with approximately, 50 query topics for each that can be used for retrieval. The same is for **TREC2013** where two sets are also available (2013 and 2014). We combined each collection query sets into one, having a total of more than hundred queries for each collection. Table 4.1 gives an overview of each collection.

[1] The microblog track in TREC is mainly concerned on tasks such as Ad-hoc search in microblog environments (Mainly Twitter).

Collection	**Number of tweets**	**Tweets collection period**	**Number of queries (*combined*)**
TREC2011	16M	16 days	108
TREC2013	243M	59 days	115

Table 4.1: Overview of collections used in our experiments.

All collections are accessible through a search API[2] provided by the microblog track organizers. The tweets retrieved through the search API are not filtered to follow TREC microblog track rules of relevant tweets. There are two main rules that would label a tweet as "non-relevant" to a query:

- non-English tweet: non-English tweets were not considered relevant in this work. Hence, a language detection tool developed by Cybozu Labs [3] is used [20]. The tool is a java library that uses Bayesian filtering to detect a text language. As reported, it gives 99% precision for 53 languages.

- Retweet[4]: retweets are also filtered out [20]. Such tweets are marked with "RT" and thus are eliminated from our retrieval results.

4.1.2 Retrieval Model and Evaluation Measures

4.1.2.1 Retrieval Model

Any retrieval task mentioned throughout this thesis uses *query likelihood model* [39]. We call the initial retrieval of a set T_q for a query q using query likelihood model the **baseline** model, where the initial set retrieved T_q is the set used for evaluation and for comparison to measure the improvement of the proposed methodology with regard to the baseline model. In all the experiments conducted, the baseline of TREC2011 is the main baseline used for evaluation comparison unless otherwise mentioned. As mentioned in section 3.2, the query likelihood model $P(D|Q)$ ranks documents by the probability that the query text could be generated by the document language model [9]. Using Bayes' rule it is calculated as the following:

$$P(D|Q) \propto P(Q|D)P(D) \tag{4.1}$$

$P(D)$ is considered uniform. Therefore it can be ignored from the equation above 4.1. $P(Q|D)$ can be calculated as the following [9]:

$$P(Q|D) = \prod_{i=1}^{n} P(q_i|D) \tag{4.2}$$

where $P(q_i|D)$ can be computed using MLE.

[2] https://github.com/lintool/twitter-tools/wiki

[3] https://code.google.com/p/language-detection/

[4] A retweet is a repost or forward of a message posted by another (user).

4.1.2.2 Evaluation Measures

Given the query set provided by TREC for the collections, we can submit each query to retrieve a ranked set of tweets. Each query in the set has a timestamp where retrieved tweets should intuitively precede it. Both test collections have a set of judgments that indicate which tweets in the collection are relevant to each query. Those judgments are used to evaluate the effectiveness of the retrieval model with regard to the retrieved set of tweets for a query. As discussed in the microblog track overview papers [28, 29, 38, 43], the primary evaluation measures used to evaluate a retrieval model are *precision at 30* (**P@30**) and *Mean Average Precision* (**MAP**). To explain such evaluation measures a basic evaluation measure must be explained first. That is *precision*. Precision represents the fraction of retrieved documents that are relevant [34] and it is calculated as follows:

$$Precision = \frac{\#\text{ relevant retrieved documents}}{\#\text{ retrieved documents}} \tag{4.3}$$

Precision at 30 (**P@30**) is calculated normally as in equation 4.3 but within the first 30 retrieved documents (tweets) only.
The Mean average precision (MAP) for a set of queries is computed as the mean of each query *average precision* (AP):

$$AP(q) = \frac{1}{|m_q|}\sum_{k=1}^{|m_q|} Precision@(R_k) \tag{4.4}$$

where $|m_q|$ is the number of relevant documents to q and R_k is the rank of the k^{th} relevant document in the results of q.

$$MAP(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} AP(q_j) \tag{4.5}$$

where $|Q|$ is the number of queries.

4.2 Research Questions and Evaluations

In order to evaluate the effectiveness of our proporsed methodology, we focused our evaluation on the following research questions:
RQ1: Which ranking function is most effective for relevant hashtags retrieval?
RQ2: Can we recommend better hashtags if we consider temporal aspects?
RQ3: How can hashtags be represented?
RQ4: How effective is query expansion with hashtags?
RQ5: How does the expansion model preform over different test collections?

4.2.1 Hashtag Retrieval and Temporal Aspects (RQ1 and RQ2)

In this section we focus on experiments that would analyze results answering the first two research questions. We have proposed three ranking methods explained earlier in section 3.5: KL, Cosine and Temporal Decay. KL and Cosine functions are the main ranking functions used, however, we incorporate a temporal aspect into the function's calculations producing four more ranking functions (explained in section 3.5.3):

- **KL_t_Recent**, uses the same KL ranking function but with temporal scoring incorporation. In addition to that, the hashtag time used in this ranking function is represented using the most recent tweet timestamp that contains the hashtag with regard to the query timestamp.
- **KL_t_Avg**, also uses KL ranking function with temporal decay but having the hashtag time represented using the average timestamps of tweets that contain the hashtag.
- **Cosine_t_Recent**, is based on the Cosine ranking fucntion but with temporal scoring incorporation and having the hashtag time equal to the most recent tweet timestamp.
- **Cosine_t_Avg**, uses Cosine ranking function with temporal decay but having the hashtag time represented using the average timestamps of tweets.

Table 4.2 gives a briefing of each ranking function:

Name	**Temporal?**	**Hashtag Time**
KL	✗	✗
Cosine	✗	✗
KL_t_Recent	✓	Most recent Tweet timestamp
KL_t_Avg	✓	Average of tweets timestamps
Cosine_t_Recent	✓	Most recent Tweet timestamp
Cosine_t_Avg	✓	Average of tweets timestamps

Table 4.2: A briefing of all ranking functions.

To determine which ranking function is most effective in hashtag retrieval, we randomly selected 30 queries from the query set of TREC2013 collection and a pool of all top 10 hashtags retrieved from all six ranking functions listed above were shared with unbiased judges for their evaluation. The total number of hashtags evaluated was 275. There were a couple of variants that we set empirically (different variants are detailed in chapter 3). The hashtags retrieved were extracted from the top 50 tweets (**h_extr**) of the original query results and top tweets to calculate terms weights in both hashtag and query model **MLE_h_tweets** and **MLE_q_tweets** were set to 40. The number of terms added to the query and hashtag vector model was set to 20.

The judges who evaluated the hashtags were 5, 4 of which have bachelor degree in computer science and the fifth is a statistical researcher. Each judge was given a set of 6 queries with an alphabetically sorted pool of retrieved hashtags were they are labeled by the judges as "relevant" or "non-relevant". Although it is preferable to include all TREC2013 queries and for them to be judged by two different judges to have maintain accuracy but this wasn't applied in this study due to the time limitations. Eventually, we used the labeled hashtags to evaluate the precision from 1 to 10 of each ranking function. Table 4.3 shows precision @ k ($p@k$), were k changes from 1 to 10 for all the 30 queries. For the detailed precision values for all the queries, refer to appendix A table A.1.

	KL	**Cosine**	**KL_t_recent**	**KL_t_Avg**	**Cosine_t_recent**	**Cosine_t_Avg**
P@1	0.4333	0.4000	0.4000	0.4000	**0.4667**	**0.4667**
P@2	0.3667	0.3500	0.3667	0.3500	**0.4167**	0.4000
P@3	0.3444	0.3000	**0.3556**	**0.3556**	**0.3556**	**0.3556**
P@4	0.3417	0.3000	0.3417	0.3417	**0.3583**	**0.3583**
P@5	0.3333	0.3067	0.3400	0.3467	**0.3533**	0.3467
P@6	0.2889	0.2833	0.2944	0.3000	**0.3056**	**0.3056**
P@7	0.2667	0.2667	0.2810	0.2857	**0.2905**	**0.2905**
P@8	0.2667	0.2708	0.2750	0.2750	**0.2958**	0.2917
P@9	0.2593	0.2667	0.2667	0.2667	0.2815	**0.2852**
P@10	0.2067	0.2200	0.2133	0.2133	0.2267	**0.2367**

Table 4.3: $p@k$ (k from 1 to 10) for retrieving "relevant" hashtags, averaged over 30 randomly-selected queries from TREC 2013 collection. Values in **bold** show the highest precision among all ranking functions.

The values in the table above 4.3 are illustrated in Figure 4.1.

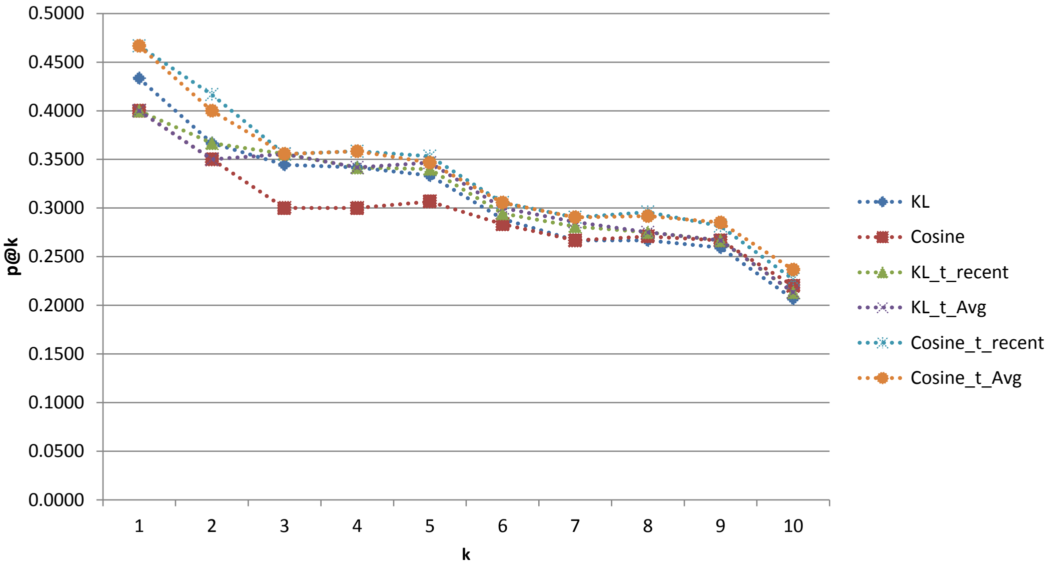
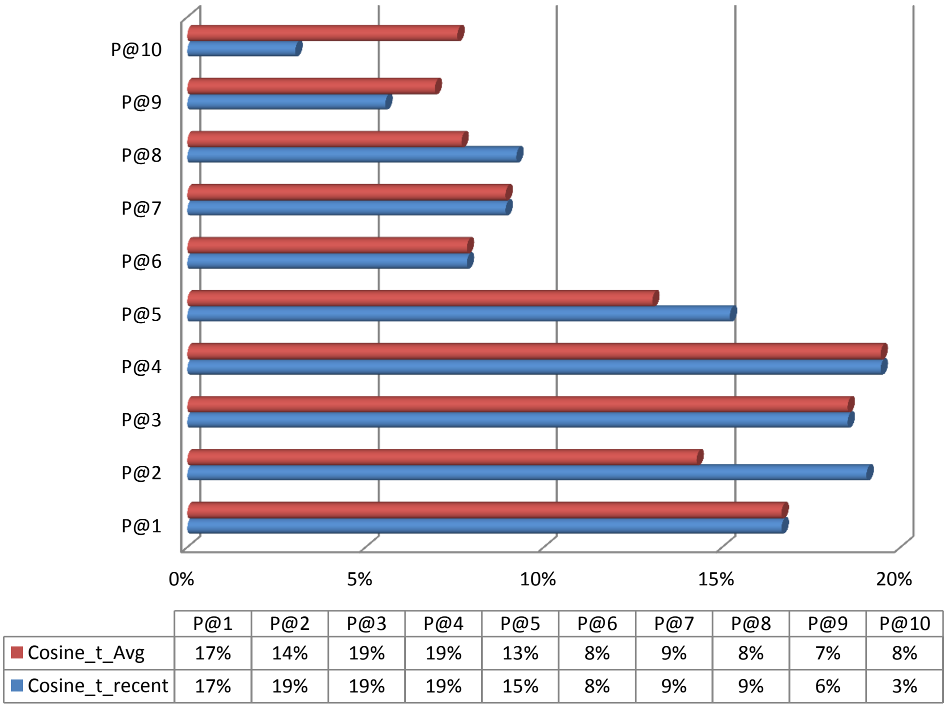


Figure 4.1: $p@k$ for each ranking function, averaged over all random queries and all ranking functions.

It is noticeable from the results above that the Cosine ranking functions with temporal decay, **Cosine_t_Recent** and **Cosine_t_Avg**, outperforms other ranking functions having **Cosine_t_Recent** function marginally higher than **Cosine_t_Avg**.

If we focus on the precision values of Cosine ranking function and compare it with the precision values when we considered temporal decay, we can confirm that all precision values increase in both temporal cosine ranking functions **Cosine_t_Recent** and **Cosine_t__Avg** by a maximum of 17% at the first recommended hashtag. Figure 4.2 gives the percentages of improvement of Cosine ranking function with temporal decay over the non-temporal Cosine ranking function.



	P@1	P@2	P@3	P@4	P@5	P@6	P@7	P@8	P@9	P@10
🟥 Cosine_t_Avg	17%	14%	19%	19%	13%	8%	9%	8%	7%	8%
🟦 Cosine_t_recent	17%	19%	19%	19%	15%	8%	9%	9%	6%	3%

Figure 4.2: Precentages of improvement for both temporal Cosine ranking functions over non-temporal Cosine function.

KL ranking function also shows an improvement using temporal decay especially when recommending more than 2 hashtags (starting from p@3 and above).

4.2.2 How Can Hashtags Be Represented?

Representing hashtags is one of the key factors that can effect a ranking function performance. The goal of this section is to study the best methodology in representing the hashtag model to have an effective match between the hashtag and the query models. The focus here is specifically on the hashtag model since the query model would have the same representation specifics as the hashtag model.

Referring to section 3.4, there are two main factors that can change the hashtag model representation:

1. The number of results (tweets) to extract hashtags from (**h_extr**).

2. The number of terms to be added to the hashtag vector representation (**vector_nTerms**).

We experimented with both variants on the largest test collection **TREC2013** that we used earlier to evaluate the best performing ranking function. **Cosine_t_Recent** is

the ranking function that we experiment with given that it is the best performing one as shown in answering the first research question in section 4.2.1. Using the ranked hashtags we use our query expansion model with one expansion hashtag to report the results and to evaluate how the approximation variants can affect the model representation and eventually the expansion model. Since we have expansion with only one hashtag, we set β to 0.2 and α to 0 were no expansion terms are used. The number of terms added to the hashtag and query vector model **vector_nTerms** was set to equal 20 in addition to the original query terms. Furthermore, referring to the approach chapter (sections 3.2 and 3.3) we experimentally selected the number of tweets to calculate the vector terms weights (**MLE_h_tweets**) and (**MLE_q_tweets**) to be 40.
Table 4.4 gives P@30 and MAP values of the expansion model using different number hashtag extraction tweets **h_extr**. From the experimented results, we can see that setting the number of tweets to be 15 gives the highest precision and setting it to 25 gives the highest MAP. During our experiments we use a value between the two and set **h_extr** to be 20.

	5	**10**	**15**	**20**	**25**	**50**	**100**
P@30	0.4622	0.4567	**0.4644**	0.4633	0.4617	0.4628	0.4578
MAP	0.2650	0.2603	0.2664	0.2664	**0.2665**	0.2629	0.2631

Table 4.4: The Average P@30 and MAP for query expansion model using different number of tweets to extract hashtags. Values in **bold** are the highest for each evaluation measure.

To experiment with the number of added terms to the hashtag vector model **vector_nTerms**, we used the same settings and fixed **h_extr** to be 20. Table 4.5 reports the different average values using different number of terms where having the number of terms to be 25 or 30 gives the best expansion results compared to lower or higher number of terms.

	5	**10**	**15**	**20**	**25**	**30**	**40**	**50**
P@30	0.4606	0.4622	0.4611	0.4633	**0.4650**	**0.4650**	0.4611	0.4606
MAP	0.2649	0.2648	0.2647	0.2664	**0.2665**	**0.2665**	0.2638	0.2623

Table 4.5: The Average P@30 and MAP for query expansion model using different number of terms to be added to the hashtags vector model. Values in **bold** are the highest for each evaluation measure.

Based on this, we set the number of terms in both hashtag and query vector model to be 25.

4.2.3 How Effective Is Query Expansion With Hashtags?

In this section, we analyze the expansion model with experiments conducted to evaluate how effective is hashtag expansion and how many hashtags the expansion is best with. In addition to that, it is empirical to test expansion terms, non-hashtag terms, and to compare it to hashtag-only expansion. Eventually, we conducted a study on combining both expansion schemes. An analysis on the query level is also conducted in this section to understand in details what expansion hashtags or terms the model retrieves for sample queries.

For the next research questions that are studied in the remainder of this chapter, values of parameters are fixed and using the best reported values in the previous research questions, unless mentioned otherwise. One of which is the main ranking function which is set to be Cosine_t_recent since it was reported to be the best performing ranking function in the first research question 4.2.1. Also, it is important to mention that **TREC2013** collection is the main collection used for the upcoming research questions except for the last research question (**RQ5**). The number of tweets used to extract hashtags **h_extr** is set to 20 and the number of terms added to both hashtag and query vector model **vector_nTerms** is equal to 25 (following research question 4.2.2) in addition to the original query terms. Furthermore, referring to the approach chapter (sections 3.2 and 3.3) we experimentally selected the number of tweets to calculate the vector terms weights (**MLE_h_tweets**) and (**MLE_q_tweets**) to be 40.

Different value settings of both expansion model parameters α and β are experimented in table 4.6. We experiment the values starting from 0.1 until 0.4 so that the sum of the parameters α and β wouldn't be larger than the original query tuning parameter $(1 - \alpha - \beta)$ (equation 3.13). As shown in the table 4.6, the best performing value for α and β is 0.1.

		β			
		0.1	**0.2**	**0.3**	**0.4**
α	**0.1**	**0.4817**	0.4728*	0.4406	0.3567
	0.2	0.4617**	0.4494	0.4067	0.3361
	0.3	0.4394	0.4200	0.3811	0.2983
	0.4	0.3989	0.3744	0.3483	0.2489

Table 4.6: The Average P@30 for different α and β values using one hashtag and one terms expansion. Value in **bold** is the highest among all values. Value with * is the second best and ** is the third.

4.2.3.1 Expansion with Hashtags Only

First of all we study expansion with multiple hashtags. For this experiment we empirically set β to 0.2 and α to 0 sine no expansion terms are used. As mentioned before (section 3.6), there are three different weighting schemes to weight the expansion hashtags in the final expanded query, **HFB1**, **HFB2** and the original ranking score **Ranking_score**. For each weighting scheme, we test expansion with one hashtag to a maximum of 30 hashtags. All expansion runs are compared to the (**baseline**) model where no expansion is used (zero hashtags) and the original query results retrieved, using query likelihood model, are the final results set that is evaluated. Following the same settings mentioned earlier in this section, Table 4.8 shows the precision results of expanding with different number of hashtags using the three different weighting scheme. The precision evaluation results show no improvement compared to the baseline system. However, expansion with 2 hashtags using HFB1 weight give better precision compared to other number of hashtags or weights. We label this expansion model as (**2H_HFB1**).

	Baseline	1	2	3	4	5	10	20	30
HFB1	**0.4728**	0.4694	**0.4728**	0.4717	0.4700	0.4639	0.4617	0.4628	0.4639
HFB2	**0.4728**	0.4650	0.4683	0.4694	0.4694	0.4689	0.4689	0.4689	0.4689
Ranking score	**0.4728**	0.4644	0.4667	0.4678	0.4667	0.4683	0.4661	0.4661	0.4661

Table 4.7: P@30 for query expansion model using different number of expansion hashtags and different hashtag weighting schemes compared to the baseline system. Values in **bold** indicate the highest average precision for each weighting scheme.

	Baseline	1	2	3	4	5	10	20	30
HFB1	0.2726	0.2646	0.2721	0.2734	0.2733	**0.2744**	0.2711	0.2714	0.2709
HFB2	**0.2726**	0.2665	0.2647	0.2653	0.2657	0.2653	0.2645	0.2646	0.2646
Ranking score	**0.2726**	0.2644	0.2692	0.2707	0.2682	0.2715	0.2697	0.2697	0.2697

Table 4.8: MAP for query expansion model using different number of expansion hashtags and different hashtag weighting schemes compared to the baseline system. Values in **bold** indicate the highest average precision for each weighting scheme.

4.2.3.2 Expansion with Non-hashtag Terms

Now we experiment the expansion model using non-hashtag expansion terms having α set to 0.2 and β to 0. Table 4.9 shows the precision results of expanding with one term up to 30 and also compared to the baseline model. As the table shows, the maximum precision and map value is when the number of terms is equal to 20 (**20T**). The percentage of improvement is 4% over the baseline with regard to precision.

	Baseline	1	2	3	4	5	10	20	30
P@30	0.4728	0.4656	0.4861	0.4800	0.4839	0.4861	0.4889	**0.4917**	0.4883
MAP	0.2726	0.2696	0.2910	0.2952	0.2958	0.2987	0.3005	**0.3018**	0.3011

Table 4.9: P@30 and MAP for query expansion model using different number of expansion (non-hashtag) terms compared to the baseline system.

We have empirically set the number of tweets to retrieve the expansion terms from to be 40.

4.2.3.3 Expansion with Both Hashtags and Terms

Given the best performing precision results from expanding with hashtags only 4.2.3.1 and with non-hashtag terms 4.2.3.2, we can combine both expansion methods and study if the combination improves the results retrieval or not. The selected expansion model using hashtags only is using **HFB1** hashtag weighting scheme and with two hashtags for expansion (**2H_HFB1**). For non-hashtag expansion terms, the best performing model was using 20 expansion terms (**20T**). Combining both expansion methods (**2H+20T**) having α and β parameters set to 0.2, gives a higher average precision value over using each expansion method separately. In addition to that, the combined expansion gives a 5% improvement in precision and 12% in MAP over the baseline system. Table 4.10 gives the P@30 and MAP of combining both expansion methods and compared to the baseline system.

	Baseline	**2H_HFB1**	**20T**	**2H+20T**
P@30	0.4728	0.4728	0.4917	**0.4972**
MAP	0.2726	0.2721	0.3018	**0.3060**

Table 4.10: The P@30 and MAP for each expansion method with hashtags and non-hashtag terms separately or combined compared to the baseline system. Value in **bold** indicates the highest value.

We also calculate the percentages for the number of improved queries, having a larger precision value, over the baseline system for each expansion method among all 2013 queries and report the results in Figure 4.3.

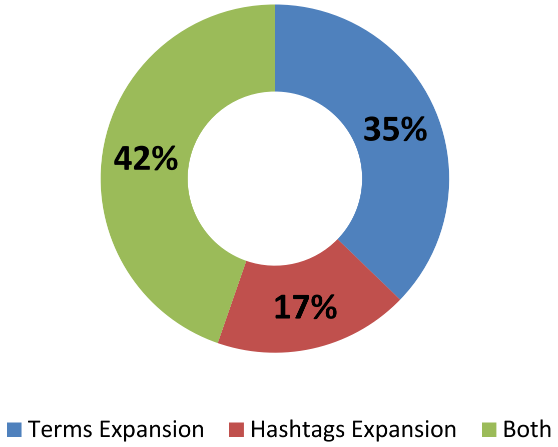


Figure 4.3: Chart illustrating the percentage of improved queries when expanding with hashtags (2H_HFB1) and non-hashtag terms (20T) separately or combined (2H+20T) compared to baseline.

The highest percentage 42% is achieved when the best expansion method is used which is expanding with both hashtags and non-hashtag terms (2H+20T). 17% of queries had a worse performance compared to the baseline in all expansion methods and the rest remained the same with no improvement.

4.2.3.4 Query Analysis

To understand how the query expansion model is performing for each query, we take a look in this section on sample queries to study what expansion hashtags and expansion terms does the model retrieve. Also, we try to analyze at the query level how the model improves the results retrieval. For this analysis we use the best performing model from the previous research questions and we analyze the results of expansion with 2 hashtags (2H_HFB1) and 20 expansion terms (20T) (section 4.2.3.3).

We select two sample queries from the judged queries pool explained in 4.2.1 to display the sample expansion hashtags and terms that are retrieved by the expansion model. Table 4.11 presents the two queries with a brief description of their subject and the list of retrieved hashtags and terms. Note that all hashtags retrieved on both sample queries were judged to be “relevant”.

Query ID	Query	Brief Description	Expansion Hashtags	Expansion Terms
MB133	cruise ship safety	talks about an event where a ship's, named “Carnival cruise”, engine room overpowered and the fire knocked out power and plumbing across most of the vessel and left it adrift in the Gulf of Mexico [2].	#breakingnews #cnn	drill five crew industry members fiv reveals secrets insider ex officer carnival canaries killed resulted death wrong gone died lifeboat
MB137	cause of the Super Bowl blackout	talks about the event where the power goes off during a super bowl game causing a blackout [1].	#nfl #houston	superdome beyonce xlvii orleans shutdown lights eyonc news column out power halftime chances freakout went twitter hurts host depressing wacky

Table 4.11: Sample queries and the retrieved expansion hashtags and terms.

If we take a look at the average precision values of both queries in Table 4.12, we can notice that different expansion techniques can differ in improving the queries.

QID	**2H_HFB1**	**20T**	**2H+20T**
MB133	**0.7667**	0.7000	**0.7667**
MB137	0.2333	**0.2667**	0.1667

Table 4.12: Sample queries MB133 and MB137 average precision values among the baseline system, 2H_HFB1, 20T, and 2H+20T.

Query MB133 has the best retrieval results when the expansion model uses hashtags only (2H_HFB1) or hashtag in addition to non-hashtags expansion (2H+20T). However, query MB137 has the highest results when the model uses terms only expansion (20T). It is important to mention that any improvement of a query using hashtags only may have an improved results when using combined expansion (2H+20T) but this is not guaranteed. To experimentally prove that, we have selected the top 13 queries that have high precision values when expanding with hashtags only (2H_HFB1) and we study the different expansion methods precision values and compare them in Figure 4.4

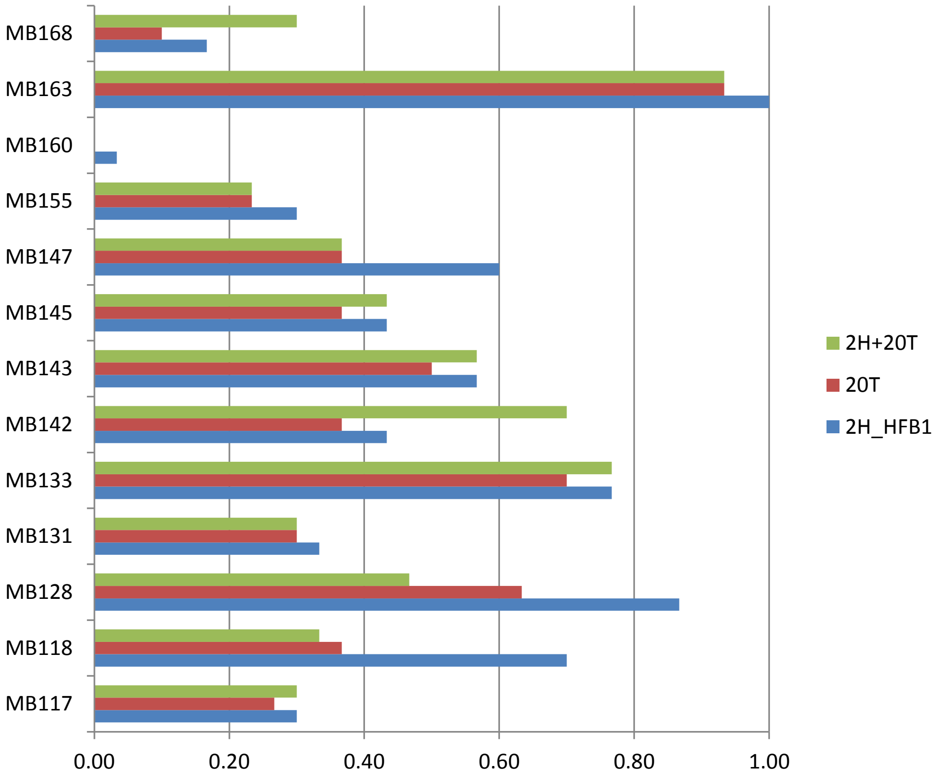


Figure 4.4: Comparison of expansion methods with the best performing queries using 2H_HFB1.

The figure shows that in most of the queries the combined expansion can improve the results but not for queries MB118 and MB128.

For all 2013 queries, 12% of the queries that improved using hashtags-only expansion (2H_HFB1) also improved with the combined expansion (2H+20T). The details of all 2013 queries precision values compared with all expansion methods can be found in the appendix B table B.1.

Referring back to table 4.12 we tried to analyze queries that had worse retrieval results when using hashtags for expansion such as query MB137. Although the query's retrieved hashtags were judged to be "relevant", the retrieval results evaluation was worse when expanding with hashtags. We believe that the hashtags are considered relevant given the time the query event happened but the retrieval method that we used in this study (query likelihood model) is non-temporal, hence could not retrieve tweets that are temporally relevant to the query nor the hashtags used for expansion. In query MB137 for example, the hashtag #houston would be relevant at the time period of the event the query time represents but not after that, and thus when the retrieval model is used without any temporal consideration, it might cause a concept drift and retrieve tweets talking about houston in general and not about the event when the NFL game was held.

4.2.4 How Does The Expansion Model Preform Over Different Test Collections?

We study what we believe to be the best performing expansion model (2H+20T) on both collections **TREC2011** and **TREC2013** and using the combined query sets (TREC11-12) and (TREC13-14) and the baselines of those combined sets. We experimented with two different values of α and β. Table 4.13 shows evaluation results having both α and β equal to 0.1 and 4.14 shows evaluation results having both α and β equal to 0.2. The tables indicate an overall slight improvement for both collections using both hashtag-only expansion (2H_HFB1) and combined expansion (2H+20T) over the Baseline.

	TREC11-12				**TREC13-14**			
	Baseline	**2H_HFB1**	**20T**	**2H+20T**	**Baseline**	**2H_HFB1**	**20T**	**2H+20T**
P@30	0.3864	0.3904	**0.4105**	**0.4105**	0.5501	0.5499	0.5530	**0.5586**
MAP	0.3007	0.3028	0.3241	**0.3255**	0.3462	0.3485	0.3712	**0.3730**

Table 4.13: Comparison of P@30 and MAP improvement compared to the baseline among both collections TREC11-12 and TREC13-14 using the expansion models (2H_HFB1, 20T and 2H+20T) having both α and β equal to 0.1. Values in **bold** are the highest for each evaluation measure and collection.

Having both α and β equal to 0.1, the percentage of precision improvement using combined expansion over the baseline is 6% and 2% for (TREC11-12) and (TREC13-14) respectively. Whereas, the MAP percentage of improvement is 8% for both collections.

	TREC11-12				**TREC13-14**			
	Baseline	**2H_HFB1**	**20T**	**2H+20T**	**Baseline**	**2H_HFB1**	**20T**	**2H+20T**
P@30	0.3864	0.3914	0.4157	**0.4194**	0.5501	0.5545	**0.5771**	0.5693
MAP	0.3007	0.3032	**0.3303**	**0.3303**	0.3462	0.3484	**0.3929**	0.3903

Table 4.14: Comparison of P@30 and MAP improvement compared to the baseline among both collections TREC11-12 and TREC13-14 using the expansion models (2H_HFB1, 20T and 2H+20T) having both α and β equal to 0.2. Values in **bold** are the highest for each evaluation measure and collection.

When α and β values are set to 0.2, the percentage of precision improvement using the same expansion method (combined expansion) over the baseline is 9% and 3% for (TREC11-12) and (TREC13-14) respectively and the MAP percentage of improvement is 10% for (TREC11-12) and 13% for (TREC13-14).

Evaluating the significance of the improvement that hashtag expansion shows was experimented by comparing different P@30 results shown in table 4.14 above using two-tailed paired t-test with significance level $p = 0.05$. The test compared the baseline with hashtag-only expansion (2H_HFB1) on one hand and on the other hand we compared terms only expansion (20T) with the combined expansion (2H+20T). None of the hashtag expansion methods experimented showed any significant improvement.

4.3 Results Comparison

As mentioned before in chapter 3, our work is based on Efron's [12] work described earlier in the background and related work chapter (section 2.2). In this section, we compare the main findings from our experiments to Efron's work. Table 4.15 below gives the main comparison points that distinguish between both studies.

Comparison	**Efron** [12]	**This Study**
Number of Collections	1	2
Collection Size	3M	16M and 243M
Number of Queries	39	223
Ranking Functions	KL	KL, Cosine Similarity, and Temporal Decay
Significance ($p < 0.05$)	Significant Improvement	Not Significant

Table 4.15: Comparison between Efron and this study by collections, number of queries, ranking functions, and significance test.

The first difference is the collections used, their size and the number of queries that were used for expirements. Efron evaluated the work on data collected using Twitter's streaming API[5]. The data was collected on a 24-hours interval having over 3 million tweets and 39 query topics. Whereas in our work, the methodology was tested on two widely-used collections in microblog research **TREC2011** and **TREC2013**. **TREC2011** has over 16 million tweets collected over a period of 16 days and has 108 queries. **TREC2013** was collected on a period of 59 days collecting over 243 million tweets and has 115 queries for experiments.

In our work, we used two main ranking functions: KL-divergence and Cosine similarity to rank recommended hashtags and we further incorporated temporal decay to both ranking functions. Efron, on the other hand, used only KL-divergence for ranking.

The main outcome of the evaluation that Efron conducted showed an improvement when using hashtags for expansion. "*All runs using hashtag-based feedback gave results that were statistically significantly better than the baseline run using standard term-based feed-back.*" (Efron [12]). We tried to test the significance of difference using P@30 using two-tailed paired t-test with $p = 0.05$. Unlike Efron, none of the hashtag expansion runs, including the combined run (2H+20T), that showed improvement in our experiments, had a significant improvement.

[5] http://api.twitter.com

Chapter 5

Conclusion and Future Work

Based on the experiments and analysis conducted and presented in Chapter 4, we summarize a set of conclusions and future work proposals detailed in the following sections.

5.1 Conclusion

This study proposes an enhanced query expansion feedback model that incorporates hashtags for expansion to improve ad-hoc search in Twitter. The study was conducted on both **TREC2011** and **TREC2013** test collections with over 100 queries for each and using P@30 and MAP as the main evaluation measures.

There are two main ranking functions experimented to retrieve hashtags that are potentially relevant to a user query: KL divergence and Cosine similarity. Temporal incorporation is also introduced in both ranking functions and with the help of user judgments, the best performing ranking function is determined to be Cosine similarity with temporal decay. Adding temporal aspect improved the Cosine similarity performance by approximately 17% on average on the first recommended hashtag.
The expansion model was used with three types of expansion methods:

- Expansion with hashtags-only.
- Expansion with non-hashtag terms.
- Expansion with the combination of both hashtags and non-hashtag terms.

The non-hashtag terms expansion model experimentally proved to perform better than the baseline system in TREC2013 collection whereas hashtags-only expansion model had the same performance as the baseline system in the best performing hashtag-only expansion scenario. The experiments showed that the combined expansion of hashtags and non-hashtag terms is the best performing expansion model which improves in precision by 5% over the baseline system and by 12% in MAP. It also improves the retrieval performance in both TREC collections and using both evaluation measures.

With a detailed analysis focusing on the query level performance, 12% of the queries that originally have improved using hashtag-only expansion were further improved using the combined expansion model.

During the background investigation and experiments conducted for this work, we managed to co-author a paper submitted for TREC 2013 microblog track and contributed in implementing query expansion using top terms [20]. We hope that we can similarly publish another paper that is based on this work's experiments to share it with the public research community.

5.2 Future Work

There are several methods proposed in the literature that can be combined and explored using our enhanced model. One is using SVM classification to classify spam hashtags or advertising ones that wouldn't improve a query results retrieval when used.

A technique of dynamically expanding queries can be studied to improve the results were each query would be expanded with the best suitable method of expansion either by hashtags-only, terms only or both hashtags and terms.

Hashtag representation can be another area to explore were a hashtag is represented in the final expanded query with a set of terms that better describe its topic. Top terms from top hashtag relevant tweets can be explored. Also, LDA seems an intriguing method for hashtag representation.

In this work, we explored two main ranking functions to recommend hashtags and there are several other functions that can be explored and further examined such as using different temporal decay incorporation functions.

Bibliography

[1] 15 things that may have caused super bowl power outage. http://mashable.com/2013/02/03/super-bowl-power-outage/. Publish Date: 2013-02-04, Accessed Date: 2015-06-16.

[2] Crippled carnival cruise ship limps into alabama. http://articles.chicagotribune.com/2013-02-14/news/chi-cruise-ship-20130214_1_cruise-ship-costa-concordia-carnival-cruise-lines. Publishing Newspaper: Chicago Tribune, Publish Date: 2013-02-14, Accessed Date: 2015-06-16.

[3] Younos Aboulnaga and Charles L Clarke. Frequent itemset mining for query expansion in microblog ad-hoc search. Technical report, DTIC Document, 2012.

[4] I. Anagnostopoulos, V. Kolias, and P. Mylonas. Socio-semantic query expansion using twitter hashtags. In *Semantic and Social Media Adaptation and Personalization (SMAP), 2012 Seventh International Workshop on*, pages 29–34, Dec 2012.

[5] Piyush Bansal, Somay Jain, and Vasudeva Varma. Towards semantic retrieval of hashtags in microblogs. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 7–8, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[6] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[7] Paul Alexandru Chirita, Claudiu S. Firan, and Wolfgang Nejdl. Personalized query expansion for the web. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 7–14, New York, NY, USA, 2007. ACM.

[8] Jaeho Choi and W. Bruce Croft. Temporal models for microblogs. *Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM '12*, page 2491, 2012.

[9] Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, USA, 1st edition, 2009.

[10] Anqi Cui, Min Zhang, Yiqun Liu, Shaoping Ma, and Kuo Zhang. Discover breaking events with popular hashtags in twitter. *Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM '12*, page 1794, 2012.

[11] Zhuoye Ding, Xipeng Qiu, Qi Zhang, and Xuanjing Huang. Learning topical translation model for microblog hashtag suggestion. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 2078–2084, 2013.

[12] Miles Efron. Hashtag retrieval in a microblogging environment. *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10*, page 787, 2010.

[13] Miles Efron and Gene Golovchinsky. Estimation methods for ranking recent information. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 495–504, New York, NY, USA, 2011. ACM.

[14] Miles Efron, E Daniel Street, Peter Organisciak, and Katrina Fenlon. Improving Retrieval of Short Texts Through Document Expansion. *SIGIR12*, (12-16 August 2012):911–920, 2012.

[15] T El-Ganainy, Z Wei, W Magdy, and W Gao. QCRI at TREC 2013 Microblog Track. *Text REtrieval Conference*, 2013, 2013.

[16] Bailan Feng, Juan Cao, Zhineng Chen, Yongdong Zhang, and Shouxun Lin. Multimodal query expansion for web video search. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 721–722, New York, NY, USA, 2010. ACM.

[17] Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 593–596, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[18] Z Han, X Li, M Yang, H Qi, S Li, and T Zhao. HIT at TREC 2012 Microblog Track. *Text REtrieval Conference*, pages 267–276, 2012.

[19] Morgan Harvey and Fabio Crestani. Long time, no tweets! time-aware personalised hashtag suggestion. In Allan Hanbury, Gabriella Kazai, Andreas Rauber, and Norbert Fuhr, editors, *Advances in Information Retrieval*, volume 9022 of *Lecture Notes in Computer Science*, pages 581–592. Springer International Publishing, 2015.

[20] Maram Hasanain, Latifa Al-marri, and Tamer Elsayed. QU at TREC-2013 : Expansion Experiments for Microblog Ad-hoc Search. *Text REtrieval Conference*, 2013.

[21] Nattiya Kanhabua and Kjetil Nørvåg. Quest: Query expansion using synonyms over time. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III*, ECML PKDD'10, pages 595–598, Berlin, Heidelberg, 2010. Springer-Verlag.

[22] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):pp. 79–86, 1951.

[23] Yin-Hsi Kuo, Kuan-Ting Chen, Chien-Hsing Chiang, and Winston H. Hsu. Query expansion for hash-based image object retrieval. In *Proceedings of the 17th ACM International Conference on Multimedia*, MM '09, pages 65–74, New York, NY, USA, 2009. ACM.

[24] Su Mon Kywe, Tuan-anh Hoang, Ee-peng Lim, and Feida Zhu. On Recommending Hashtags in Twitter Networks. In *Proceedings of the 4th international conference on Social Informatics*, pages 337–350, 2012.

[25] Cheng Li, Yue Wang, and Qiaozhu Mei. A Double-Loop Process for Investigational Search : Foreseers in TREC 2013 Microblog Track. *Text REtrieval Conference*, 2013.

[26] Feng Liang, Runwei Qiang, and Jianwu Yang. PKU_ICST at TREC 2011 Microblog Track. *Text REtrieval Conference*, 2011.

[27] Feng Liang, Runwei Qiang, and Jianwu Yang. Exploiting Real-Time Information Retrieval in the Microblogosphere. *Joint Conference on Digital Library (JCDL)*, 2012.

[28] Jimmy Lin and Miles Efron. Overview of the TREC-2013 Microblog Track (Notebook Draft), 2013.

[29] Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. Overview of the TREC-2014 Microblog Track (Notebook Draft), 2014.

[30] Xiaoyong Liu and W. Bruce Croft. Cluster-based retrieval using language models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 186–193, New York, NY, USA, 2004. ACM.

[31] Samuel Louvan, Mochamad Ibrahim, and Mirna Adriani. University of Indonesia at TREC 2011 microblog track. *Text REtrieval Conference*, 2011.

[32] Zhiqiang Ma, Wenwen Dou, Xiaoyu Wang, and Srinivas Akella. Tag-latent dirichlet allocation: Understanding hashtags and their relationships. *Proceedings - 2013 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2013*, 1:260–267, 2013.

[33] Zongyang Ma, Aixin Sun, and Gao Cong. Will this #hashtag be popular tomorrow? *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '12*, page 1173, 2012.

[34] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[35] Taiki Miyanishi, Kazuhiro Seki, and Kuniaki Uehara. Improving pseudo-relevance feedback via tweet selection. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*, pages 439–448, 2013.

[36] Apostol (Paul) Natsev, Alexander Haubold, Jelena Tešić, Lexing Xie, and Rong Yan. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *Proceedings of the 15th International Conference on Multimedia*, MULTIMEDIA '07, pages 991–1000, New York, NY, USA, 2007. ACM.

[37] Eriko Otsuka, Scott a. Wallace, and David Chiu. Design and evaluation of a Twitter hashtag recommendation system. *Proceedings of the 18th International Database Engineering & Applications Symposium on - IDEAS '14*, pages 330–333, 2014.

[38] Iadh Ounis, Craig Macdonald, Jimmy Lin, and Ian Soboroff. Overview of the TREC-2011 Microblog Track, 2011.

[39] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

[40] Runwei Qiang, Yue Fei, Yihong Hong, and Jianwu Yang. PKUICST at TREC 2013 Microblog Track. *Text REtrieval Conference*, 2013, 2013.

[41] Sheikh Muhammad Sarwar, Md. Anowarul Abedin, A. H. M. Sofi Ullah, and Abdullah Al Mamun. Personalized query expansion for web search using social keywords. In *Proceedings of International Conference on Information Integration and Web-based Applications &#38; Services*, IIWAS '13, pages 610:610–610:614, New York, NY, USA, 2013. ACM.

[42] Ian Soboroff, Dean McCullough, Jimmy Lin, Craig Macdonald, Iadh Ounis, and Richard McCreadie. Evaluating real-time search over tweets. *Proc. ICWSM*, pages 943–961, 2012.

[43] Ian Soboroff, Iadh Ounis, and Jimmy Lin. Overview of the TREC-2012 microblog track, 2012.

[44] Jaime Teevan, Daniel Ramage, and Merredith Ringel Morris. #twittersearch: A comparison of microblog search and web search. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 35–44, New York, NY, USA, 2011. ACM.

[45] Hayley Tsukayama. Twitter turns 7: Users send over 400 million tweets per day. http://articles.washingtonpost.com/2013-03-21/business/37889387_1_tweets-jack-dorsey-twitter, 2013.

[46] Shao-Chi Wang and Yuzuru Tanaka. Topic-oriented query expansion for web search. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 1029–1030, New York, NY, USA, 2006. ACM.

[47] Jianshu Weng, Ee-Peng Lim, Qi He, and Cane Wing-Ki Leung. What Do People Want in Microblogs? Measuring Interestingness of Hashtags in Twitter. *2010 IEEE International Conference on Data Mining*, pages 1121–1126, December 2010.

[48] Stewart Whiting, Iraklis A. Klampanos, and Joemon M. Jose. Temporal pseudo-relevance feedback in microblog retrieval. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7224 LNCS, pages 522–526, 2012.

[49] Feng Xiao, Tomoya Noro, and Takehiro Tokuda. News-topic oriented hashtag recommendation in twitter based on characteristic co-occurrence word detection. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7387 LNCS, pages 16–30, 2012.

[50] Eva Zangerle, Wolfgang Gassler, and Gunther Specht. Recommending#-tags in twitter. In *Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings*, volume 730, pages 67–78, 2011.

[51] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 334–342, New York, NY, USA, 2001. ACM.

Appendix A

A.1 Judgments Evaluation for Ranking Functions

To determine which ranking function performs best in hashtag retrieval, hashtags are labeled by judges as “relevant” or “non-relevant”. Eventually, labeled hashtags are used to evaluate the precision from 1 to 10 of each ranking function. Table A.1 below shows precision values from 1 to 10 for all randomly selected queries along with the total average of all precisions for each ranking function.

Query	KL Precision	Cosine Precision	KL_t_recent Precision	KL_t_Avg Precision	Cosine_t_recent Precision	Cosine_t_Avg Precision	Avg
MB111: water shortages	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.5000	0.5000	0.1667
	0.0000	0.0000	0.0000	0.0000	0.3333	0.3333	0.1111
	0.0000	0.2500	0.0000	0.0000	0.2500	0.2500	0.1250
	0.0000	0.2000	0.0000	0.0000	0.2000	0.2000	0.1000
	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667
	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429
	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250
	0.1111	0.1111	0.1111	0.1111	0.1111	0.1111	0.1111
	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
MB113: Kal Penn	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MB115: memories of Mr. Rogers	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MB117: marshmallow Peeps dioramas	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	0.5000	0.5000	1.0000	1.0000	1.0000	1.0000	0.8333
	0.6667	0.3333	0.6667	0.6667	1.0000	1.0000	0.7222
	0.7500	0.2500	0.5000	0.5000	1.0000	1.0000	0.6667
	0.6000	0.4000	0.6000	0.6000	1.0000	1.0000	0.7000
	0.5000	0.5000	0.5000	0.5000	0.8333	0.8333	0.6111
	0.4286	0.4286	0.4286	0.4286	0.7143	0.8571	0.5476
	0.3750	0.3750	0.3750	0.3750	0.7500	0.7500	0.5000
	0.3333	0.3333	0.3333	0.3333	0.7778	0.7778	0.4815
	0.3000	0.3000	0.3000	0.3000	0.7000	0.8000	0.4500
MB119: colony collapse disorder	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000	0.3333

	0.0000	0.5000	0.0000	0.0000	0.5000	0.5000	0.2500
	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333
	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500
	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667
	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429
	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250
	0.1111	0.1111	0.1111	0.1111	0.1111	0.1111	0.1111
	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
MB121: Future of MOOCs	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667
	0.5000	0.5000	0.7500	0.7500	0.5000	0.5000	0.5833
	0.4000	0.4000	0.6000	0.8000	0.4000	0.4000	0.5000
	0.5000	0.5000	0.6667	0.8333	0.5000	0.5000	0.5833
	0.5714	0.5714	0.5714	0.7143	0.5714	0.5714	0.5952
	0.5000	0.6250	0.6250	0.6250	0.6250	0.6250	0.6042
	0.5556	0.6667	0.6667	0.6667	0.6667	0.6667	0.6481
	0.6000	0.6000	0.7000	0.7000	0.6000	0.7000	0.6500
MB123: solar flare	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.3333	0.0000	0.3333	0.3333	0.0000	0.0000	0.1667
	0.2500	0.0000	0.2500	0.2500	0.0000	0.0000	0.1250
	0.2000	0.0000	0.2000	0.2000	0.2000	0.2000	0.1667
	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667
	0.1429	0.1429	0.1429	0.1429	0.2857	0.2857	0.1905
	0.1250	0.1250	0.1250	0.1250	0.2500	0.2500	0.1667
	0.2222	0.2222	0.2222	0.2222	0.2222	0.2222	0.2222
	0.2000	0.2000	0.2000	0.2000	0.3000	0.3000	0.2333
MB125: Oscars snub Affleck	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

	0.0000	0.0000	0.0000	0.0000	0.3333	0.3333	0.1111
	0.2500	0.2500	0.2500	0.2500	0.5000	0.5000	0.3333
	0.4000	0.4000	0.4000	0.4000	0.6000	0.6000	0.4667
	0.5000	0.5000	0.5000	0.5000	0.6667	0.6667	0.5556
	0.5714	0.5714	0.5714	0.5714	0.7143	0.7143	0.6190
	0.6250	0.5000	0.6250	0.6250	0.7500	0.7500	0.6458
	0.6667	0.5556	0.6667	0.6667	0.6667	0.7778	0.6667
	0.7000	0.6000	0.7000	0.7000	0.6000	0.7000	0.6667
MB127: Hagel nomination filibustered	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333
	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000
	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667
	0.7143	0.7143	0.7143	0.7143	0.7143	0.7143	0.7143
	0.7500	0.7500	0.7500	0.7500	0.7500	0.7500	0.7500
	0.7778	0.7778	0.7778	0.7778	0.7778	0.7778	0.7778
MB129: Angry Birds cartoon	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	0.8333	0.8333	0.9444
	0.8571	1.0000	1.0000	1.0000	0.8571	0.8571	0.9286
	0.8750	1.0000	0.8750	0.8750	0.8750	0.8750	0.8958
	0.8889	0.8889	0.8889	0.8889	0.8889	0.8889	0.8889
	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000
MB131: trash the dress	1.0000	0.0000	1.0000	1.0000	1.0000	0.0000	0.6667
	0.5000	0.0000	0.5000	0.5000	0.5000	0.0000	0.3333
	0.3333	0.0000	0.3333	0.3333	0.3333	0.3333	0.2778

	0.2500	0.0000	0.5000	0.5000	0.5000	0.5000	0.3750
	0.2000	0.2000	0.4000	0.4000	0.6000	0.4000	0.3667
	0.3333	0.3333	0.5000	0.5000	0.5000	0.5000	0.4444
	0.2857	0.2857	0.4286	0.4286	0.5714	0.5714	0.4286
	0.3750	0.3750	0.3750	0.3750	0.5000	0.5000	0.4167
	0.4444	0.4444	0.4444	0.4444	0.5556	0.5556	0.4815
	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
MB133: cruise ship safety	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	0.6667	0.6667	0.8889
	1.0000	0.7500	1.0000	1.0000	0.7500	0.7500	0.8750
	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000
MB135: Big Dog terminator robot	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.3333	0.0000	0.0000	0.3333	0.3333	0.1667
	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500
	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
	0.3333	0.1667	0.1667	0.1667	0.1667	0.1667	0.1944
	0.2857	0.1429	0.2857	0.2857	0.1429	0.1429	0.2143
	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500
	0.2222	0.2222	0.2222	0.2222	0.2222	0.2222	0.2222
MB137: cause of the Super Bowl blackout	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	0.5000	1.0000	0.5000	0.5000	1.0000	1.0000	0.7500
	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667
	0.7500	0.7500	0.7500	0.7500	0.7500	0.7500	0.7500
	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000
	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667
	0.7143	0.5714	0.7143	0.7143	0.5714	0.5714	0.6429
	0.7500	0.6250	0.7500	0.7500	0.6250	0.6250	0.6875

	0.6667	0.6667	0.7778	0.7778	0.6667	0.6667	0.7037
	0.7000	0.7000	0.8000	0.8000	0.7000	0.7000	0.7333
MB139: Artists Against Fracking	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	0.5000	0.5000	0.5000	0.5000	0.6667
	0.6667	0.6667	0.6667	0.6667	0.3333	0.3333	0.5556
	0.5000	0.5000	0.5000	0.5000	0.2500	0.2500	0.4167
	0.4000	0.4000	0.4000	0.4000	0.2000	0.2000	0.3333
	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333
	0.2857	0.2857	0.2857	0.2857	0.2857	0.2857	0.2857
	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500
	0.2222	0.2222	0.2222	0.2222	0.2222	0.2222	0.2222
	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
MB141: Mila Kunis in Oz movie	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MB143: Maracana Stadium problems	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.1667	0.1667	0.0556
	0.0000	0.0000	0.0000	0.0000	0.1429	0.1429	0.0476
	0.0000	0.1250	0.0000	0.0000	0.1250	0.1250	0.0625
	0.0000	0.1111	0.0000	0.0000	0.1111	0.1111	0.0556

	0.0000	0.1000	0.0000	0.0000	0.1000	0.1000	0.0500
MB145: National Parks sequestered	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2500	0.0000	0.2500	0.2500	0.0000	0.0000	0.1250
	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
	0.1667	0.1667	0.1667	0.1667	0.3333	0.3333	0.2222
	0.2857	0.1429	0.2857	0.2857	0.2857	0.2857	0.2619
	0.3750	0.2500	0.3750	0.3750	0.3750	0.3750	0.3542
	0.4444	0.3333	0.4444	0.4444	0.3333	0.3333	0.3889
	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
MB147: Victoria's Secret commercial	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.3333
	0.5000	0.5000	0.5000	0.5000	0.0000	0.0000	0.3333
	0.3333	0.3333	0.3333	0.3333	0.0000	0.0000	0.2222
	0.2500	0.2500	0.2500	0.2500	0.0000	0.0000	0.1667
	0.2000	0.2000	0.2000	0.2000	0.0000	0.0000	0.1333
	0.1667	0.1667	0.1667	0.1667	0.0000	0.0000	0.1111
	0.1429	0.1429	0.1429	0.1429	0.0000	0.0000	0.0952
	0.1250	0.1250	0.1250	0.1250	0.0000	0.0000	0.0833
	0.1111	0.1111	0.1111	0.1111	0.1111	0.1111	0.1111
	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
MB149: making football safer	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.1429	0.1429	0.0476
	0.0000	0.0000	0.0000	0.0000	0.1250	0.1250	0.0417

	0.0000	0.0000	0.0000	0.0000	0.1111	0.1111	0.0370
	0.0000	0.0000	0.0000	0.0000	0.1000	0.1000	0.0333
MB151: gun advocates are corrupt	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MB153: lighter bail for Pistorius	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000	0.3333
	0.0000	0.0000	0.0000	0.0000	0.5000	0.5000	0.1667
	0.0000	0.0000	0.0000	0.0000	0.3333	0.3333	0.1111
	0.0000	0.0000	0.0000	0.0000	0.2500	0.2500	0.0833
	0.2000	0.0000	0.2000	0.2000	0.2000	0.2000	0.1667
	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667
MB155: Obama reaction to Syrian chemical weapons	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.5000	0.0000	0.0000	0.5000	0.5000	0.2500
	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333
	0.2500	0.5000	0.2500	0.2500	0.5000	0.5000	0.3750
	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
MB157: Kardashian maternity style	1.0000	1.0000	1.0000	1.0000	0.0000	1.0000	0.8333
	1.0000	0.5000	1.0000	1.0000	0.5000	0.5000	0.7500
	0.6667	0.6667	0.6667	0.6667	0.3333	0.3333	0.5556
	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000

	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333
	0.2857	0.2857	0.2857	0.2857	0.4286	0.2857	0.3095
	0.2500	0.2500	0.2500	0.2500	0.3750	0.2500	0.2708
	0.2222	0.2222	0.2222	0.2222	0.3333	0.2222	0.2407
	0.2000	0.2000	0.2000	0.2000	0.3000	0.3000	0.2333
MB159: circular economy initiatives	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	0.8333
	1.0000	0.0000	1.0000	1.0000	0.5000	0.5000	0.6667
	1.0000	0.3333	1.0000	1.0000	0.3333	0.3333	0.6667
	1.0000	0.5000	0.7500	0.7500	0.5000	0.5000	0.6667
	1.0000	0.6000	0.8000	0.8000	0.4000	0.4000	0.6667
	0.8333	0.6667	0.8333	0.8333	0.5000	0.5000	0.6944
	0.7143	0.7143	0.7143	0.7143	0.4286	0.4286	0.6190
	0.6250	0.6250	0.6250	0.6250	0.3750	0.3750	0.5417
	0.6667	0.6667	0.6667	0.6667	0.4444	0.4444	0.5926
	0.6000	0.7000	0.6000	0.6000	0.5000	0.4000	0.5667
MB161: 3D printing for science	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.2500	0.0000	0.0000	0.2500	0.2500	0.1250
	0.2000	0.2000	0.2000	0.2000	0.4000	0.4000	0.2667
	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333
MB163: virtual currencies regulation	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	0.5000	0.5000	0.5000	0.5000	1.0000	1.0000	0.6667
	0.3333	0.3333	0.6667	0.6667	0.6667	0.6667	0.5556
	0.5000	0.2500	0.5000	0.5000	0.5000	0.5000	0.4583
	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
	0.3333	0.5000	0.3333	0.3333	0.3333	0.3333	0.3611
	0.2857	0.4286	0.4286	0.4286	0.4286	0.4286	0.4048
	0.2500	0.3750	0.3750	0.3750	0.5000	0.5000	0.3958

	0.2222	0.4444	0.3333	0.3333	0.4444	0.5556	0.3889
	0.3000	0.5000	0.3000	0.3000	0.4000	0.5000	0.3833
MB165: ACPT Crossword Tournament	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	0.5000	1.0000	1.0000	0.9167
	0.6667	0.6667	0.6667	0.6667	1.0000	1.0000	0.7778
	0.5000	0.7500	0.5000	0.5000	0.7500	0.7500	0.6250
	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000
	0.5000	0.5000	0.5000	0.5000	0.6667	0.6667	0.5556
	0.5714	0.5714	0.5714	0.5714	0.5714	0.5714	0.5714
	0.6250	0.6250	0.6250	0.6250	0.6250	0.6250	0.6250
	0.5556	0.5556	0.5556	0.5556	0.5556	0.5556	0.5556
MB167: sequestration opinions	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	0.6667	0.6667	0.6667	0.6667	1.0000	1.0000	0.7778
	0.5000	0.5000	0.5000	0.5000	0.7500	0.7500	0.5833
	0.4000	0.4000	0.4000	0.4000	0.6000	0.6000	0.4667
	0.3333	0.3333	0.3333	0.3333	0.5000	0.5000	0.3889
	0.2857	0.4286	0.2857	0.2857	0.4286	0.4286	0.3571
	0.3750	0.3750	0.3750	0.3750	0.3750	0.3750	0.3750
MB169: Honey Boo Boo Girl Scout cookies	0.0000	1.0000	0.0000	0.0000	1.0000	1.0000	0.5000
	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333
	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500
	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667
	0.2857	0.2857	0.2857	0.2857	0.1429	0.1429	0.2381
	0.2500	0.2500	0.2500	0.2500	0.1250	0.1250	0.2083
	0.3333	0.3333	0.2222	0.2222	0.1111	0.1111	0.2222

	0.3000	0.4000	0.3000	0.3000	0.2000	0.2000	0.2833

Table A.1: Precision for top 10 retrieved hashtags for all random judged queries.

Appendix B

B.1 Evaluation of Expansion Methods for all 2013 Collection Queries

The details of all 2013 collection queries precision values compared with all expansion methods are shown in the following table:

QID	Query	Baseline	2H_HFB1	20T	2H+20T
MB111	water shortages	0.1667	0.1667	0.1667	0.1667
MB112	Florida Derby 2013	0.3333	0.3333	0.3667	0.3333
MB113	Kal Penn	0.3000	0.2667	0.2667	0.3000
MB114	Detroit EFM Undemocratic	0.1667	0.1667	0.1667	0.1667
MB115	memories of Mr. Rogers	0.3667	0.3667	0.3667	0.3667
MB116	Chinese Computer Attacks	0.7000	0.7000	0.8000	0.8333
MB117	marshmallow Peeps dioramas	0.3000	0.3000	0.2667	0.3000
MB118	Israel and Turkey reconcile	0.5667	0.7000	0.3667	0.3333
MB119	colony collapse disorder	0.4667	0.4667	0.5000	0.5000
MB120	Argentina's Inflation	0.4000	0.4000	0.4000	0.4000
MB121	Future of MOOCs	0.7667	0.6333	0.8000	0.7667
MB122	unsuccessful kickstarter applicants	0.0000	0.0000	0.0000	0.0000
MB123	solar flare	0.5333	0.5333	0.9000	0.9000
MB124	celebrity DUI	0.0333	0.0333	0.0333	0.0333
MB125	Oscars snub Affleck	0.9000	0.9000	0.9333	0.9333
MB126	Pitbull rapper	1.0000	1.0000	1.0000	1.0000
MB127	Hagel nomination filibustered	0.9333	0.9333	0.9333	0.9667
MB128	Buying clothes online	0.8667	0.8667	0.6333	0.4667
MB129	Angry Birds cartoon	0.9333	0.9333	1.0000	0.9667
MB130	Lawyer jokes	0.2667	0.2667	0.2667	0.2667
MB131	trash the dress	0.4667	0.3333	0.3000	0.3000
MB132	asteroid hits Russia	0.1000	0.1000	0.1667	0.2000
MB133	cruise ship safety	0.7333	0.7667	0.7000	0.7667
MB134	The Middle TV show	0.1000	0.0667	0.0667	0.0333
MB135	Big Dog terminator robot	0.3333	0.3333	0.3333	0.2667
MB136	Gone Girl reviews	0.1000	0.1000	0.1000	0.1000

MB137	cause of the Super Bowl blackout	0.2667	0.2333	0.2667	0.1667
MB138	New York City soda ban blocked	1.0000	1.0000	1.0000	1.0000
MB139	Artists Against Fracking	0.6333	0.6333	0.6333	0.6333
MB140	Richard III burial dispute	0.7000	0.7000	0.7667	0.7667
MB141	Mila Kunis in Oz movie	1.0000	1.0000	1.0000	1.0000
MB142	Iranian weapons to Syria	0.3667	0.4333	0.3667	0.7000
MB143	Maracana Stadium problems	0.5333	0.5667	0.5000	0.5667
MB144	Downton Abbey actor turnover	0.5000	0.4333	0.6667	0.6333
MB145	National Parks sequestered	0.3667	0.4333	0.3667	0.4333
MB146	GMO labeling	0.9667	0.9667	0.9667	0.9667
MB147	Victoria's Secret commercial	0.6000	0.6000	0.3667	0.3667
MB148	Cyprus Bailout Protests	0.2667	0.2333	0.5000	0.5667
MB149	making football safer	0.7000	0.7000	0.7000	0.7000
MB150	UK wine industry	0.0000	0.0000	0.0000	0.0000
MB151	gun advocates are corrupt	0.0000	0.0000	0.0000	0.0000
MB152	Iceland FBI Wikileaks	0.7333	0.7333	0.7333	0.7333
MB153	lighter bail for Pistorius	0.1667	0.1667	0.1667	0.1667
MB154	anti-aging resveratrol	0.5000	0.4667	0.6333	0.6333
MB155	Obama reaction to Syrian chemical weapons	0.2333	0.3000	0.2333	0.2333
MB156	Bush's dog dies	0.9000	0.9000	0.9667	0.9667
MB157	Kardashian maternity style	0.9667	0.9667	1.0000	1.0000
MB158	hush puppies meal	0.1333	0.1667	0.2667	0.3000
MB159	circular economy initiatives	0.6000	0.6333	0.6333	0.6333
MB160	social media as educational tool	0.0333	0.0333	0.0000	0.0000
MB161	3D printing for science	0.3667	0.2333	0.5333	0.5000
MB162	DPRK Nuclear Test	0.3667	0.3667	0.3667	0.3667
MB163	virtual currencies regulation	0.8667	1.0000	0.9333	0.9333
MB164	Lindsey Vonn sidelined	0.2667	0.2667	0.5333	0.5333
MB165	ACPT Crossword Tournament	0.0667	0.0333	0.0667	0.0333
MB166	Maryland casino table games	0.3667	0.3667	0.3667	0.3667
MB167	sequestration opinions	0.1667	0.1667	0.2000	0.2000
MB168	US behind Chaevez cancer	0.1000	0.1667	0.1000	0.3000
MB169	Honey Boo Boo Girl Scout cookies	1.0000	1.0000	1.0000	0.9333
MB170	Tony Mendez	0.8000	0.8000	0.8333	0.8333
Average		0.4728	0.4728	0.4917	0.4972

Table B.1: Precision values for all 2013 queries for baseline, Hashtag only expansion (H2_HFB1), non-hashtag terms expansion (20T) and both expansion shcemes combined (2H+20T).