TREC Microblog Track 2012
Real-Time Adhoc Retrieval

Aayush Bhandari, CSC849
aayushb@sfsu.edu

1 INTRODUCTION

Tweets or Microblogs are a great source to study the modern Internet society, including changes in various trending events. Microblogs can be used to understand complex pattern in human behaviors and can foster many scientific studies. The tweet data's are readily available for TREC[1] participants and makes approaching these studies a lot easier. However tweets come from all over the world and talk about many different topics. Hence, it is important to have an approach to examine only the relevant tweets for any given topic.

The *Microblog Track* was first introduced to TREC in 2011. The track involves using twitter [2]data to retrieve the most relevant tweets for a given query, based on the information need. The tweets are retrieved from a pre-compiled corpus rather than a live stream. Hence to honor time sensitiveness, a time stamp is required to be submitted as a part of the query. This allows only the relevant and recent tweets around that time period to be retrieved as a part of the search.

My interest in the microblog track arises because of twitters real time nature and the fact that the microblog track has been part of TREC for a few years now. With respect to concurrency, analyzing the most recent tweets can be beneficial in recognizing the world's reaction to a certain topic. Many organizations are constantly looking for the means to get this information for their concerned research areas. Hence the project is still concurrent and could contribute to improvements in the retrieval of relevant tweets. The other reason, the track being part of TREC, allows the access to relevance judgments for various topics from the previous years. This would make it very easy to analyze and adjust several design features of our system's implementation.

2 RELATED WORK

TREC 2011 had several contributors who participated in the Microblog Track and received very good results for the relevance of the retrieved tweets. Zhongyuan et al. [1] utilized query expansion using relevance feedback model to enhance the original query. Kim et al. [2] also implemented query expansion using pseudo relevance feedback and concluded that it improved their results. Another approach both of these work used was document expansion. Tweets are very short and can only contain 140 characters. Zhongyuan et al [1] saw improvements in results after applying Document Expansion Language Model to improve the representation of short tweets. On the other hand Kim et al. [2] decided to expand their document using headlines of articles usually pointed by the tweets. In both cases the documents length was increased for having more terms allowing statistical estimations to be much more reliable.

URL's in the tweets played a major role in determining the representation of the documents in the corpus. Information retrieved from these URL's was used to add important information missing from a document. Feng et al. [3] tried document expansion by selecting the top-k TF-IDF that results in very poor words such as usernames and misspelled words to be selected. Hence, the work relied on extracting titles or descriptions from the html documents, rather than doing any mathematical estimation. All of

[1] http://trec.nist.gov/
[2] https://twitter.com/

these work involved several steps for cleaning the tweets including deleting re-tweets and non-English tweets. Indri[3] seemed to be popular tool used for the purposes of indexing and retrieving relevant results.

3 PROPOSED WORK

The two different tasks in the TREC microblog track were real-time adhoc retrieval and the real-time filtering. For the proposed project I plan to implement a real-time adhoc retrieval system for twitter data. The goal is to retrieve the most recent and relevant tweets for a given query and time stamp without using any future evidence.

3.1 Pre Processing

The collection documents that contain the tweets for this project are in json format. Some of the objects in the json file are tweets, tweet-id, re-tweet status and time the tweet was generated. Before performing any indexing it is very crucial to perform several pre-processing to get the documents ready.
I have planned to write a custom parser that converts any json notation into xml so that Indri could be used to index the xml files. The parser would use the Boolean status of re-tweet to disregard any of the tweets that have the value of re-tweeted as true. The second step would be determining if any given tweet is of the English language. TREC has determined any non-English tweets to be irrelevant. Hence, the parser would use Google language-detection to check the tweet for how likely it is an English tweet. Any tweet that is probably not English would be disregarded by the parser. While generating the xml documents it would probably be wise to store all the json objects as the xml tags, so that we could indicate fields for indexing.
After filtering the documents, the next important step would be expanding the document. Tweets are short and contain a maximum of 140 characters. This makes it very difficult to analyze tweets relevance at the collection level. Hence, if any tweet contains a URL to a website, a web crawler can be used to extract the titles and descriptions from the html document. This extracted information can be stored as separated field in the xml document. This can allow us to analyze the differences caused while using or not using the documents expansion during retrieval.

3.2 Indexing

At this point the documents should be ready to be indexed, for which I will be using Indri. Indexes shall contain all the fields extracted from json file and also the field for document expansion. I will be using krovetz stemmer for stemming out morphological variants of a word. I will be filtering out the stop words, since tweets are searched mostly for trending events, which would be nouns, verbs and adjectives.
For the real-time adhoc retrieval we are not allowed to use any future evidences. If all the documents were indexed into one inverted index, the collection statistics would be faulty. It would be dependent on all the tweets of the collection, before and after the time stamp for a provided topic. Hence creating individual indexes based on the topics provided by TREC would be a better solution. TREC has provided 60 different topic queries for evaluating the retrieval system. Each of the queries contains a query time and a query tweet time. The query tweet time can be used to create an index, which only contain the tweets that were tweeted before that time. This makes sure that the collection statistics is not biased of any future evidence. Hence, I will be creating 60 indexes for 60 different topics.

[3] http://www.lemurproject.org/indri/

3.3 Query Expansion and Retrieval

I will want to use pseudo relevance feedback for query expansion using the top ranked feedback documents for the original query. Using the training query I shall be tuning in the parameter to check how the results improve based on the number of terms used in expanding the query. During retrieval, results will be obtained from either restricting or allowing document expansion. The number of terms used during query expansion will also be tuned to compare the retrieval results. After the training query is ready, similar methodologies would be applied to rest of the query topics to retrieve the final results.

3.4 Evaluation

Evaluation will be done based on the available relevance judgments from TREC website. A TREC eval program with the relevance judgment file would be used to generate the evaluation results. I will be using precision at 30 and Mean Average Precision (MAP) metrics to evaluate the effectiveness of my system.

4 EXPERIMENT METHODOLOGY

4.1 DATA PRE-PROCESSING

With its character limitation of 140, twitter has allowed users to become very sophisticated at sharing their information in a very concise manner. Old techniques such as URL's and new techniques such as hash tags makes sharing information very trivial. However these styles cause the tweets to become harder to decipher. Hence, to avoid noisy data I have planned to pre-process the tweets before indexing them.

4.1.1 Document Processing

The initial tweets were received in blocks of json files, where each of the blocks represented a day. The tweets covered over a period of 2 weeks from the 24th January 2011 until 8th February. Each of the tweets was represented as a json object and contained several metadata including text of the tweet, retweet status, username and an id based on the time when the tweet was generated. For the context of my system I decided to use the text, retweet status and id of the tweet. These fields were extracted from the tweet and saved in a pseudo xml file of the type "trectext" as shown in figure 1.

```
<DOC>
    <DOCNO>{tweet_id}</DOCNO>
    <TEXT>{tweet_text} </TEXT>
</DOC>
```

Fig 1: Representation of a tweet in pseudo xml file

The 2012 microblog provided the participants with 60 different topics. Each query provided a query tweet time, when the search engine would run to retrieve the tweets. This meant that any information generated after that time could not be used for retrieval purposes. This included all the tweets generated after the query tweet time. Hence to honor the requirement of not using future evidence, 60 different pseudo xml files were created, each containing tweets that were generated before the query tweet time. As tweet id's and query tweet time were both ISO form of real time, it was very easy to compare these values and filter out id's that were greater than the time of the query generation.

4.1.2 URL Inclusion and Exclusion

When investigating numerous tweets it came to my notice that many of the URL included in the tweets contained no information regarding the tweet itself. For example any tweet that described a video and posted a YouTube link would contain no textual information in its URL. Such URL could have a negative effect on the tweets relevance score, especially when stemmers are used. Hence, I decided to create 60 different pseudo xml files, which removes the URL's from the text while keeping a track of all the tweets that originally contained the tweet in a map file. The URL's were detected using regular expression pattern in java as shown in figure 2.

```
"(?:^|[\\W])((ht|f)tp(s?):\\/\\/|www\\.)"
+ "(([\\w\\-]+\\.){1,}?([\\w\\-.~]+\\/?)*"
+ "[\\p{Alnum}.,%_=?&#\\-+()\\[\\]\\*$~@!:/{};']*)"
```

Fig 2: Regular Expression used for extracting URL's

4.1.3 Retweet and Language Filter

Before the tweets were added to the pseudo xml files, some extra filtering was done as per the TREC requirement. TREC stated in its rules that any tweets that were retweets or non-English were non relevant. Hence, I parsed the retweet status of a tweet from the json object, and excluded any tweets that had retweet as true. Google's Language Detection tool was deployed in java for the purposes of language detection. The tool recognized English in most of the right instances, and performed very well. The only time the tool failed was when the tweets were very short, usually less than 5 words and had those words as proper nouns. The high effectiveness of the tool accounts for the very few English tweets that may have been categorized as non-English.

4.1.4 Statistics

Table 1 shows that I extracted about 32 percent of the tweets from the json files as potential relevant tweets. These tweets were the only tweets used for indexing purposes. One of the most important factors to notice in table 1 is the number of missing tweets in the corpus. About 4 million tweets have been deleted, and that could be expected as this project is being done at the end of 2014, but using tweets from 2011. The tweet download tool provided by TREC does download only the currently available data and nothing from the tweets that have been deleted or user accounts that have been suspended.

	Tweet Count
Tweets Utilized for Indexing (Processed)	**3739557**
Re-Tweets (Raw)	**1079186**
Non-English Tweets (Raw)	**7515902**
Tweets in Corpus (Raw)	**11812966**
Excepted Tweets in Corpus (Raw)	**16000000**

Table 1: Statistics of the raw and processed corpuses

4.2 INDEXING

Indri was very famous in 2012 Microblog track and did very well for most participants. Hence I decided to use Indri for indexing the tweets. Indexing was done using "trectext" format for the pseudo xml files I created.

Four different types of indexes were designed to index the tweets. There were 60 indexes for each index type to account for each of the 60 provided query. This was done since having only one index would cause statistic factor such as IDF to use future evidence. The first index type was created with no specifications using just the raw query. The second index type took into consideration stemming for which Krovetz Stemmer was used. The first two index types included the URL, however the last index didn't use URL for indexing and utilized Krovetz as a stemmer.

4.3 RETRIEVAL

Before using the selected features for the retrieval it was important to find out the type of index most suitable for retrieval of the tweets. A training set was created from the first 5 queries provided by TREC. The 5 queries provided were cleaned according to the rules used to generate each type of index. The first sets of queries were raw queries and both the second and third set of queries was stemmed and also had their stop words removed. After the queries were ran against the indexes it was seen that using Krovetz stemming by removing the URL's did better than previous two indexes. Hence, I decided to use the third index to add the remaining features as described below.

4.3.1 Pseudo Relevance Feedback

I decided to perform a query expansion using Pseudo Relevance Feedback (PRF) as one of the features in the best performing index (Krovetz stemming and URL removed). Indri provides a feature for PRF using the Lavrenko's Relevance Model [2]. In this model the query is expanded using the top x documents with a highest weight for the top terms. Indri also allows the changing of the weight for the original query vs. the expanded query. When tuning the parameters for the training set the best results occurred when using 0.2 weights for the original query and 0.8 for the expanded query. Kim et al. [2] had also used PRF and performed best when using 0.2 and 0.8 as the tuning parameters.

4.3.2 Enhancing URL

During the pre-processing of the json files I stored a mapping file, which contained all tweet ids for the tweets that contained an URL. A presence of URL usually suggests more information than the plain tweet text. For example for the query "British Government Cuts", it would be very helpful to go to blogs or longer posts by people rather than just reading the 140 characters. Hence, assuming URL's increase tweets relevance, I decided to increase the relevance scores of URL containing tweets by a certain factor. After tuning in the value to the training data set, the factor of 1.15 seemed the best for increasing tweets relevance. Since the scores generated by Indri are negative, each of the tweets that contained URL's was divided by a factor of 1.12.

5 EVALUATION and RESULTS

The relevance judgment of 2012 categorized tweets into 3 possible categories of relevance: not relevant, relevant and highly relevant. A relevance judgment file was available to analyze the performance of our systems .We can see the results of the retrieval from the 3 indexes I created for all relevant tweets in table 2, and only highly relevant tweets in table 3.

Criteria	P@30	MAP
Raw	0.1747	0.0781
Krovetz	0.2729	0.1484
Krovetz + Remove Link	0.3271	0.1727

Table 2: P@30 and MAP for the three types of index created for all relevant tweets

Criteria	P@30	MAP
Raw	0.0833	0.0626
Krovetz	0.1412	0.1121
Krovetz + Remove Link	0.1616	0.1299

Table 3: P@30 and MAP for the three types of index created for only highly relevant tweets

Both the precision at 30 and MAP metric performed better for the third type of index with Krovetz indexing and URL's removed. Table 4 and Table 5 shows the results after the features were added to best scoring index for all relevant and highly relevant tweets respectively.

Feature	P@30	MAP
None	0.3271	0.1727
PRF	0.3695 (+0.046)	0.2166 (+0.044)
PRFU	0.3751 (+0.005)	0.2089 (-0.007)

Table 4: P@30 and MAP for the Pseudo Relevance Feedback and Pseudo Relevance Feedback with URL Enhanced for all relevant tweets

Feature	P@30	MAP
None	0.1616	0.1299
PRF	0.1915 (+0.030)	0.1653 (+0.035)
PRFU	0.2090 (+0.018)	0.1733 (+0.008)

Table 5: P@30 and MAP for the Pseudo Relevance Feedback and Pseudo Relevance Feedback with URL Enhanced for only highly relevant tweets

The Pseudo Relevance Feedback for top 5 documents with 10 terms significantly increased the precision of the result. The URL Enhancer was also effective in making the precessions better. The median for the precision at 30 in the TREC track for highly relevant tweet was 0.1904, which is 0.0186 lower than my final precision at 30.

Query Level Analysis

Figure 3 shows the different queries and the results for precision at 30. Two of the best performing queries were query 66 "Journalists' treatment in Egypt" and query 86 "Joanna Yeates murder". The first topic was very popular in 2011 and would easily retrieve lots of results as there is bound to be lots of tweets about the topic.

The second topic on the other hand is very specific and would probably not contain in a non-relevant tweet. On the other hand query 58 “FDA approval of drugs” and query 69 “High taxes” are very broad. The unpopularity of the tweets could result in less relevant tweets being available and the query difficulty because of topic broadness makes retrieving the desired results much harder. Hence the system needs to consider adding a feature that can narrow down the results for topics that are broad and less popular.

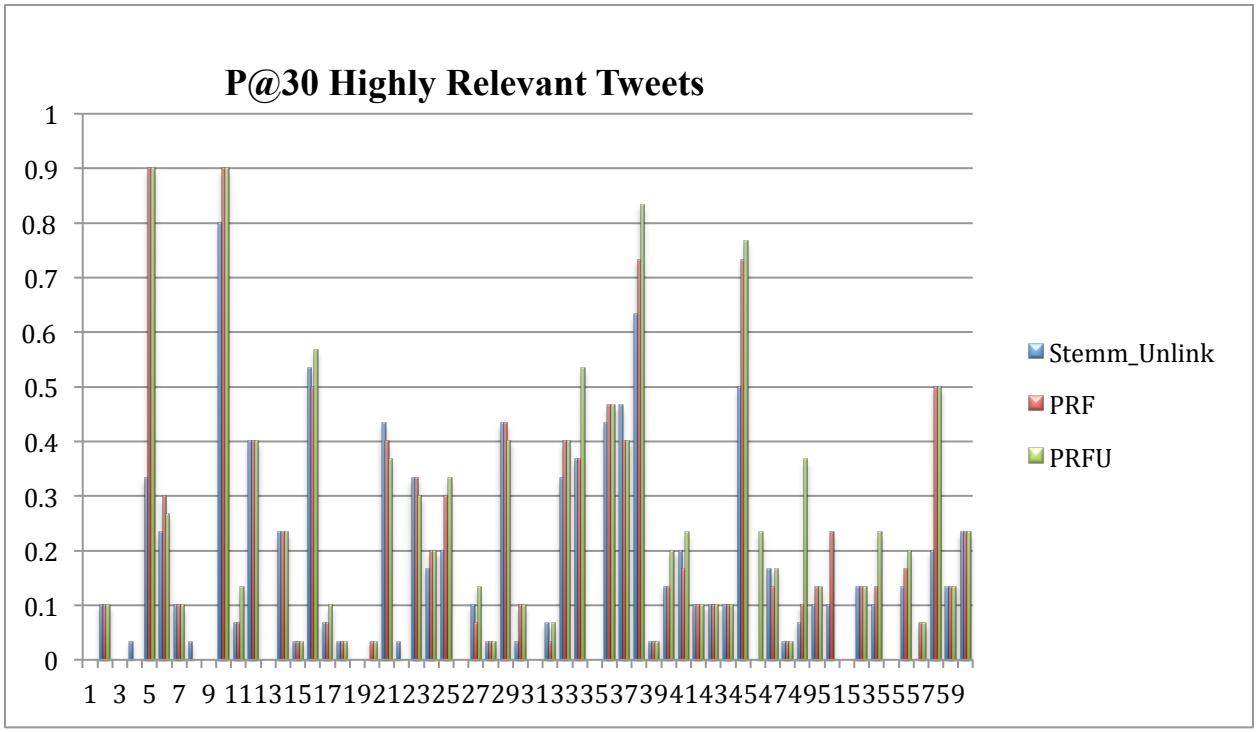

Fig 3: Precision at 30 for all 60 queries for Krovetz and URL removed Index, Pseudo Relevance Feedback Feature and Pseudo Relevance Feedback with added URL Enhance

FUTRUE WORK and CONCLUSION

One of the key features that I was unable to finish due to time limitations was document expansion from the URL contained in the tweet. This requires web crawling and could consume a significant amount of time. Many participants have tried the document expansion feature and seen a lot of improvement. It would be wise to add this feature in the future version to make the results better.

In this way I was able to download raw tweets, index them and was able to retrieve them for a given query. Most importantly, this was done without using any future evidence and based on the “query time” timestamp given for the query. I also concluded that using pseudo relevance feedback for query expansion and using URL’s presence for enhancing tweet scores could significantly make the precision of the results better.

REFERENCES

[1] Han, Z., Li, X., Yang, M., Qi, H., Li, S. & Zhao, . (2012). HIT at TREC 2012 Microblog Track. Retrieved from http://trec.nist.gov/pubs/trec21/papers/HIT_MTLAB.microblog.final.pdf

[2] Kim, Y., Teniterzi, R. & Callan, J. (2012). Overcoming Vocabulary Limitations in Twitter Microblogs. Retrieved from http://trec.nist.gov/pubs/trec21/papers/CMU_Callan.microblog.final.pdf

[3] Liang, F., Qiang, R., Hong, Y., Fei, Y. & Yang, J. (2012). PKUICST at TREC 2012 Microblog Track. Retrieved from http://trec.nist.gov/pubs/trec21/papers/PKUICST.microblog.nb.pdf