BJUT at TREC 2013 Microblog Track

Zhen YANG, Guangyuan ZHANG, Shuyong SI, Yingxu LAI, Kefeng FAN
{yangzhen, laiyingxu}@bjut.edu.cn, {zhangguangyuan, sishuyong}@emails.bjut.edu.cn, kefengfan@163.com
College of Computer Science, Beijing University of Technology, Beijing 100124, China

***Abstract*—This paper describes the first participation of BJUT in the TREC Micro-blog Track 2013. We perform the experiments on the 2013 TREC Microblog data using the standard retrieval model with several different query expansion methods including frequency method, *C* measure and Entropy differences. Also we introduce the details of our system, which consists of data preprocessing, retrieval structure, and query expansion & results analysis module.**

***Index Terms*—Microblog retrieval, Information expansion, Frequency method, *C* measure, Entropy differences**

I. Introduction

This paper describes the first participation of BJUT in the TREC Micro-blog Track 2013 [1]. This year's track focus on one single task: Real-time Adhoc Task. All participants should answer a query by providing a list of relevant tweets ranked in decreasing order by predicted relevance score.

The primary difference of the 2013 TREC from the 2011-2012 Micro-blog tracks lies in the data size, and the data size is more than 100 G. Unfortunately we cannot obtain the complete data collection, we can only receive the 10 thousand tweets at most for each topic through the official API. Therefore in this track, we focus on the query expansion and re-ranking methods for the received tweets through the API. We perform the experiments on the 2013 TREC Micro-blog data using the standard retrieval model with different query expansion methods including frequency method, *C* measure and Entropy differences.

II. Corpus and System

A. Corpus

The corpus of 2013 Micro-blog track is more than an order of magnitude larger than the previously use tweets 2011 collection. Approximately, the corpus consists of 259,057,269 tweets over a two-month period: 1 February, 2013-31 March, 2013 (inclusive). We cannot obtain the whole collection through the official API and only receive 10 thousand relevant tweets for each topic through the official search API [2]. The 10 thousand tweets are encoded by json and composed of tweet id, user id, text and so on. Then we deal with these tweets of one topic with our system to get the result.

B. Pre-Processing

There are three tasks to be done when we deal with each topic file in our system: 1) extracting the tweet id and text, 2) removing the repetitive tweets, and removing the no-English characters, and 3) the http links, removing stop-word.

Firstly, we extract the tweet id and its text of each tweet from the corpus file in one topic. If a tweet without the label "RT" in the tweets text is found, it is converted to hash format. Otherwise it is removed from data for already existing. Finally we write the content with hash format in a new file. At the same time, we delete the no-English characters, web links and stop-words of each tweet.

C. Retrieval Model

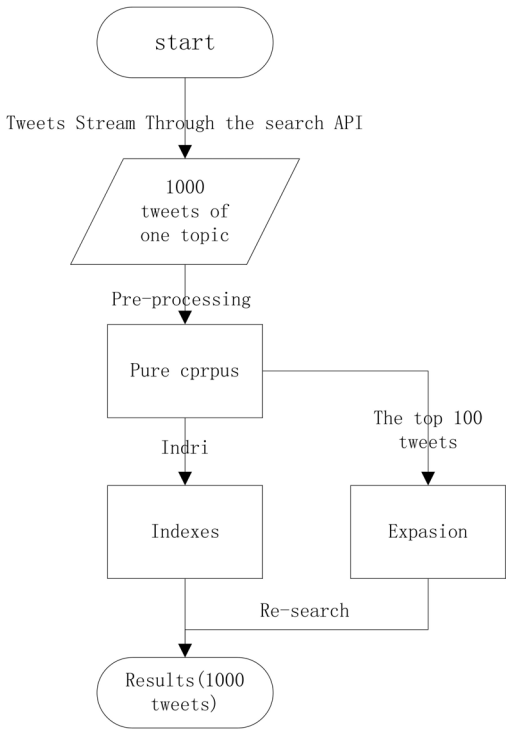

FIG. 1 The Framework of Microblog Retrieval

As shown in FIG. 1, we performed the experiments on the 2013 TREC Micro-blog data using the standard retrieval model which consists of data preprocessing, query expansion and results analysis module [3].

As we cannot get the whole collection, we only deal with each topic one by one through our system. When we get the final results, the tweets of different topics are independent of each other. And our system mainly contains three parts as the Fig.1 shows.

- The first part is corpus and preprocessing. We already detailed introduce this part in previous chapter.
- The second part is query expansion. It is the most important part. First of all, we make use of the top 100 tweets in the new corpus file of one topic to get the expansion words through our methods that will be introduced detailed in next chapter. Then we combine the original query words with the expansion words to refine search with the following formula:
  The new query words = the original query words + α (the expansion words).
- The final part is getting the results. The number of result tweets is 1000 by one topic, and one result contains the topic number, an unused column, a tweet id, the rank, the score and the run tag such as 'MB111 Q0 311248486941200386 1 -4.13049 BJUTFreq'. And we set the score with tf-idf model.

III. Query Expansion

In our system, we use three keywords extraction measures for query expansion, which are respectively based on the frequency, words' spatial distributions along the text and the theory of Shannon's entropy difference between the intrinsic and extrinsic modes which refer to the fact that relevant words significantly reflect the author's writing intentions.

For every topic we import the top 100 twitters which are pre-processed through our system into one file which we consider as an initial whole text. All of our three algorithms are based on above hypothesis.

A. Frequency Measure

This frequency based method is set as baseline. In our system, we count every word which appears in the corpus, and build a map structure to save statistical data. We filtered the stop words in our corpus by using an English stop words list. After above all works have done, we arrange all the words in reverse chronological order and select the top-5 words as the expanded key query words.

B. C Measure

P. Carpenas et al. [4] suggest using the statistical analysis of spatial distributions, i.e. spectra, to detect relevant words with the same occurrence frequency. They claim that long-range correlation or clustering (self-attraction to each other) in the spatial distribution of relevant keywords is an important feature of human-written texts, in spite of random occurrences of irrelevant words. Normalized standard deviation (C) of the nearest neighbor spacing is used to characterize the spatial distribution of a particular word.

$$C(\sigma_{nor}, n) = \frac{\sigma_{nor} - <\sigma_{nor}>(n)}{sd(\sigma_{nor})(n)} \quad (1)$$

Where $\sigma_{nor} = \frac{\sigma}{\sqrt{1-p}}$ and the parameter σ is defined as $\sigma = s / <d>$ with $<d>$ being the average distance and $s=\sqrt{<d^2> - <d>^2}$.

The C score is our purpose value. C= 0 indicates that the word appears at randomly, $C > 0$ that the word is clustered, and $C<0$ that the word repels itself. In addition, two words with the same C value can have different clustering, but the same statistical significance.

C. Entropy Differences Measure

Yang et al. [5] propose a new metric 'Entropy difference' to evaluate and rank the relevance of words in a text. The method uses the Shannon's entropy difference between the intrinsic and extrinsic mode, which refers to the fact that relevant words significantly reflect the author's writing intentions, i.e., their occurrences are modulated by the author's purposes, while the irrelevant words are distributed randomly in the text.

The idea of intrinsic-extrinsic mode is based on the general idea that relevant words are clustered, and therefore the set of distances between consecutive appearances of a word should consist of small intra-cluster distances and large inter-cluster distances.

A simple way is suggested to distinguish the intrinsic and extrinsic mode, the positions of the word occurrences in a text with frequency m are denoted by $t_1, t_2, t_3, \ldots t_m$. The distance between two successive occurrences of a word, can be written as $d_i=t_{i+1}-t_i$. The arrival time differences belongs to the intrinsic mode d^A if $d_i \leqq \mu$. Thus the intrinsic modes entropy of a word is defined as:

$$H(d^A) = -\sum_{d \in d^A} P_d \log_2 P_d \quad (2)$$

Let $d^B = \{d_i \mid d_i > \mu\}$ be the union set for all $d_i > \mu$. We can define the extrinsic mode entropy of a word as

$$H(d^B) = -\sum_{d \in d^B} P_d \log_2 P_d \quad (3)$$

Thus the entropy differences between the intrinsic and extrinsic mode can be define as follows:

$$ED^q(d) = (H(d^A))^q - (H(d^B))^q \quad (4)$$

Obviously, a word with different p randomly placed in a text would have a different entropy difference between the intrinsic and extrinsic mode, the ED_{geo}, the normalized entropy difference measure ED_{nor} is defined as

$$ED^q_{nor}(d) = ED^q(d) / |ED^q_{geo}(d)| \quad (4)$$

When we have calculated the ED_{nor} for every word, we can sort words by using it. A word with larger ED_{nor} explains that it plays more important role in the text.

IV. Results

In this year's TREC Microblog Track, we submit 3 versions of runs which are shown in the Tab. 1.

TABLE 1. RESULTS OF OUR TEAM

Run id	MAP	R-Prec	bpref	P@30	Methods
BJUTFreq	0.1088	0.1610	0.1891	0.2328	Frequency
BJUTCnor	0.0729	0.1137	0.1431	0.1822	C measure
BJUTEntr	0.0731	0.1174	0.1493	0.1639	Entropy Differences

V. CONCLUSION

In this paper, we describe the details of our methods and system structure. In our system the first method have received the best performance, the other two methods have no better performances than the first one. We think the reasons to have those results may be because we don't have the whole corpus. If we have the integrated corpus, we believe that the other two algorithms will have better performances than the first one. Furthermore, we will research the short texts to find more effective modes to describe it.

REFERENCES

[1] 2013 Micro-blog Track Guidelines, https://github.com/lintool/twitter-tools/wiki/TREC-2013-Track-Guidelines, 2013

[2] TREC 2013 API Specifications, https://github.com/lintool/twitter-tools/wiki/TREC-2013-API-Specifications, 2013.

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

[4] P. Carpena, P. Bernaola-Galván, M. Hackenberg. Level statistics of words: Finding keywords in literary texts and symbolic sequences, Physical Review E, 79(3), 2009.

[5] Zhen Yang, Jian-Jun Lei, Ke-Feng Fan, Ying-Xu Lai. Keyword extraction by entropy difference between the intrinsic and extrinsic mode. Physics A, 392(19), 4523-4531, 2013.