PRIS at TREC 2013 Microblog Track

Siming Zhu, Zhe Gao, Yajing Yuan, Hui Wang, Guang Chen
School of Information and Communication Engineering
Beijing University of Posts and Telecommunications

Abstract:

This paper described the real-time search system we built for TREC 2013 microblog track. We focused on query expansion and ranking algorithm and employed different strategies. For query expansion, we implied pseudo-relevance feedback using WAF algorithms and a refined $tf * idf$ formula. For re-ranking part, our system makes use of various tweets' features, such as expansion terms, URL information, and incorporate them in a learning–to-rank framework to improve the final ranking results.

1 Introduction

Relevance and recency are important factors in real-time Twitter search, which aims at addressing a search task whereby a user's information need is represented by a query at a specific time. This year's track consists of only one single task: real-time ad hoc search. The primary difference this year from the 2011-2012 microblog tracks lies in the tweet collection and the way that participants will interact with it.

2 Method

2.1 system overview

The system we built for real-time search task is shown in Figure 1. We dealt with Tweet2013 corpus and topics in parallel. Firstly, we downloaded the corpus remotely via a search API and did the preprocessing work. A corpus of webpage, whose links are provided in tweets are fetched by a self-designed crawler. Then we built index of tweet corpus and webpage corpus respectively, using Lemur IR toolkit. As for the topics, we used two methods for query expansion. Finally, we used a simple but effective learning to rank model which combines useful features of tweets, and re-sorted the tweets according to their relevance scores.

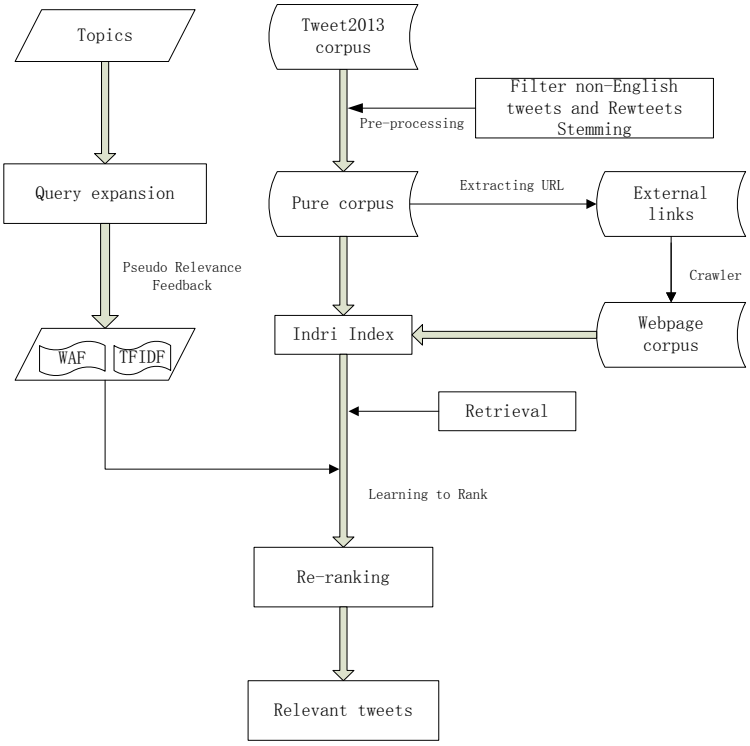

Figure 1 The framework of ad hoc search system

2.2 dataset and preprocessing

For TREC 2013, the collection consists of approximately 240 million tweets over a two-month period: 1 February, 2013 - 31 March, 2013 (inclusive). We get the official collection through the search API and downloaded each topic's top 10,000 tweets as our original corpus.

Due to the limited length of the tweet text, it fails to provide adequate information. We downloaded the URL links extracted from tweets to obtain external evidence. The total number of tweets with one or more URLs was 178,982.

In the preprocessing step, we performed two ways as following:

- **Retweets removal** .Some tweets with the sign of 'RT' are regarded as retweets, we eliminated the information after RT and kept the non-RT part.
- **Non-English tweets removal.** Microblogs are multi-lingual, as all topics are expressed in English, non-English tweets will be judged non-relevant, we use a Language-Detection to filter non-English tweets.

2.3 Query Expansion

In this stage, we are expected to mine the words that have strong connection with a given topic so as to improve document retrieval performance with more adequate information. Two algorithms were applied in this stage: the Word Activation Force algorithm and $tf * idf$ method. Both of the two methods use pseudo-relevance feedback

approach to make the most of local resources.

2.3.1 Word Activation Force Algorithm

The Word Activation Force algorithm (WAF) is based on the assumption that there's a special force in documents helping human brains activate associates of a word, such as 'papers' activates strongly 'articles' or 'letters'. It believes that there are latent structures of word network in documents. The WAF proposes an effective approach mapping syntactical and semantic information into sparse directed networks, comprehensively highlighting the features of individual word. Based on the directed networks, sensible word clusters and hierarchies can be efficiently discovered.

We used pseudo-relevance feedback approach, assuming that top-ranked tweets retrieved by API to be relevant. Thus we regarded the text of top-ranked tweets as the basic set to do query expansion.

Then words occurrence and co-occurrence were calculated in the basic set. We use the follow annotations:

- f_i ,the frequency of word i in the basic set;
- f_{ij} ,the co-occurrence of word i to word j in the basic set, which indicates the frequencies of pairs (i,j) where i precedes j by up to L words($L=4$ in our study);
- d_{ij}, the average word distance between word i and word j.

Then the word activation force of word i to word j, or waf_{ij}, can be calculated as follows:

$$waf_{ij} = \frac{\left(\frac{f_{ij}}{f_i}\right)\left(\frac{f_{ij}}{f_j}\right)}{d_{ij}^2} \quad (1)$$

We identify that the statistic is defined in the same form of the universal gravitation.

It is obvious that all the element values in the WAF matrix is between 0 and 1. Zero means that word i is never followed by word j within our word window in the basic set, while one means that word i and j are always adjacent like a compound($f_{ij} = f_j = f_i$, $d_{ij} = 1$)

With the WAF Matrix above, we can calculate the closeness of word i and j, namely affinity, as follows:

$$A_{ij}^{waf} = \left[\frac{1}{|K_{ij}|}\sum_{k\in K_{ij}} OR\left(waf_{ki}, waf_{kj}\right) \cdot \frac{1}{|L_{ij}|}\sum_{l\in L_{ij}} OR\left(waf_{il}, waf_{jl}\right)\right]^{1/2} \quad (2)$$

where $K_{ij} = \{k|waf_{ki} > 0 \ or \ waf_{kj} > 0\}$ and $L_{ij} = \{l|waf_{il} > 0 \ or \ waf_{jl} > 0\}$. And $OR(x,y) = min(x,y)/max(x,y)$. The Affinity Matrix enables us to discover the association between words in the basic set.

We calculated the Affinity Matrix of basic set, and returned top-scored words that associate to the topic word, assuming that high relevant words would have larger affinity value.

2.3.2 $Tf * idf$ method

Besides WAF algorithm, we implemented an equation that measures each term's wei ghting score.

$$Weight(t) = idf(T) * \sum_{d \in D(K)} score(d) * tf(d,T) \quad (3)$$

Where $D(K)$ is the collection of top-K tweets retrieved by API search. $idf(T)$ is the term's inverse document frequency in the whole collection. $Tf\ (d,T)$ is the term frequency that occur in the tweet, and $score(d)$ is that tweet's score.

2.4 Scoring and Ranking

Due to the limited length of the tweet text, it fails to provide adequate information. The previous research shows that whether containing URL is an important feature for a tweet. Besides, the expand words which are closely associated with the topic may contain some key information. In our ranking method, we considered both these factors.

2.4.1 Ranking Model

To rank the relevance, we use the learning to rank technique, which was successfully used in TREC 2011&2012 Microblog Track. We designed a simple linear model to combine features extracted from tweets. Given a query Q and a tweet D, the relevance $score(Q,D)$ can be computed as follows:

$$s(Q,D) = \sum_i^N \lambda_i f_i(Q,D) \quad (4)$$

where N is the number of the features, and λ_i is the coefficient of each feature.

2.4.2 Feature Extraction

The tweets we downloaded from the search API are in JSON format, which contains various features to extract. Based on previous study, we carefully analyzed the structure of microblog, and divided the features into three parts: Text_Feature, Non_text_Feature and Author_Feature.

Table 1. Features of tweets

Text_Feature	Author_Feature	Non_text_Feature
Text_score(Q,D)	Followers_Count(D)	T_diff(Q,D)
Length(D)	Retweeted_count(D)	Has_hashtag(D)
Oov_pct(D)	Statuses_count(D)	Has_url(D)
Stopwd_ratio(D)	Friends_count(D)	URL_score(Q,D)
Expanded_ratio(Q,D)	Listed_count(D)	

Based on the features' importance and the original information that tweets can provide, we chose the Text_score(Q,D), Expanded_ratio(Q,D) and URL_score(Q,D) in the Microblog Track.

- Text_score(Q,D): Besides the relevance score provided by API, which uses Lucene's implementation of query likelihood , we also use Lemur IR toolkit Indri to build index by field. And then we use both the API search relevance scores

and Indri query scores.

- Expanded_ratio(Q,D): It depends on the the number of extension words appeared in the tweet and the total number of extension words.
- URL_score(Q,D): As the URL in the microblog is an important feature, we crawled the URL pages which appeared in the tweets and build the index of webpages' titles and contents. Then we got the normalized relevance score.

We used the 2011 and 2012 dataset as the train set. To get the maximum value of P@30,we found that the number of extension words should be about 10.

2.5 Results Submission

In this year's TREC Microblog Track, we submitted 4 versions of runs:

Table 2. Four runs our team submitted

Run_Id	Text_score	Expanded_ratio	URL_score
PrisRun1	API	Yes(WAF)	No
PrisRun2	Indri	Yes(WAF)	No
PrisRun3	API	Yes(WAF)	Yes
PrisRun4	API	Yes(tf*idf)	Yes