An Information Retrieval System for FIRE 2016 Microblog Track

Trishnendu Ghorai
Department of CST
IIEST, Shibpur

ABSTRACT

This paper describes our approaches to FIRE (Forum for Information Retrieval Evaluation) 2016 Microblog track. The main aim of this track was to develop an information retrieval system that can identify relevant tweets posted during a disaster event. The relevance is measured with respect to some predefined topics provide by the track organizers. In this working note we have given the description of the system which has taken part in this year's FIRE track as well as has analysed the performance of the system.

Keywords

FIRE; information retrieval; tweet; relevancy;

1 INTRODUCTION

User written informal microblogs, like tweets, are quite important and a big source of real time information. As this microblogs are quite informal and doesn't obey standard vocabulary, thus special information retrieval system and recommendation systems are needed to retrieve information from this microblogs. To boost the retrieval performance of information retrieval system FIRE has introduced this track this year [4]. In this task the participant IR systems have to find relevant tweets from a set of tweets posted during the recent disaster time. The initial dataset consists of around 50,000 tweets from twitter that were posted in a recent Nepal earthquake. The relevancy of the tweet is measured with respect to topics, which will identify different resources that are available or required during the disaster time. The organizers provide a set of seven topics in the standard TREC format. The main challenge of the task is to tackle the nosiness of the tweets and at the same time find most relevant tweets. To deal with the problem of noise we have applied a preprocessing phase on tweets which will remove all noisy data from the tweets. The tweets are converted to a bag of words to ease up the scoring process. To calculate relevance, we have developed two different scoring and raking methods. The topics are optimized by constructing new queries based on the previous topics.

2 SYSTEM OVERVIEW

In this section we have described the system architecture for the data challenge. The system consists of tweet preprocessing, query generation, scoring of tweets and result analysis.

2.1 Brief Overview

In this task, a set of previously collected tweets (more specifically tweet ids) on Nepal Earthquake 2015 was provided. And alongside 7 queries were given in the traditional TREC format (an XML like format). The goal of this task was to find most relevant tweets from the set of tweets based on the queries.

Our system has mainly four components as follows,

1) **Tweet Preprocessing** – As tweets are informally written, tweets generally contain a lot of noise and unnecessary data. For this reason, in preprocessing stage data filters are applied on the tweets to get rid of the unwanted data.
2) **Query Construction** – The topics are provided have three parts, namely tittle, narration and description. To get more relevant tweets, a new set of queries is constructed from this given topic.
3) **Scoring of tweets** – Once the queries are constructed each tweet are scored based on each query. Two different approaches have been used in scoring the tweets.
4) **Final filtering** – When each tweet gets a score against each topic, a heuristic threshold has been set to get good quality tweets.

2.2 Tweet Preprocessing

The following steps have been taken to preprocess the tweet text.

1) **Punctuation removal** – Punctuations are removed from each tweet. We have not given any extra importance to hash tags, all '#' symbols are also removed.
2) **Case folding** – All the capital letters in the tweets are converted to small letters
3) **Stop word removal –** All commonly used English words which do not have much significance on the subject matter of the tweet but are used only for semantic reasons are removed. A list of top most frequently used words (around 500 words) are used as stop word list. And from the tweet the words that are present in the stop word list are removed.
4) **Non ASCII character –** In addition, we have removed all non ASCII characters which come to tweet due to the use of emoticons and other symbol
5) **Constructing bag of word** – Each tweet is then splited into words and are converted to a set of words. Each set represents the collection of the distinct word that are present in the tweet. Each bag of word is identified by the tweet id of the tweet which is unique to the tweet and can be used to track it in next steps.

2.3 Query Construction

Topics are made of three fields, namely the title, description and narratives. Titles contain several three or four keys, while descriptions are one-sentence long statements of the users' information needs; narratives are paragraph-length descriptions of the tweets that the users want to receive and are the long description. Each topic is assigned one topic id which can be used to uniquely specify one topic in submission stage. Query construction part consists of two different phases described as follows:

1) **Keyword Extraction** – As nouns in a sentence holds most of the information, we choose nouns in the topics as the keywords for the query. We have used Stanford Part-Of-Speech Tagger[1] to label different parts-of-speech first and then collected words which have been identified as Noun.
2) **Giving weight to keywords** – As all the topics can be broadly classified into two groups based on if it wants to retrieve tweets on 'availability' or 'requirement'. For this reason, the words like 'availability' or 'requirement' have been assigned more weight than the other key words in the topics.

Each query can be expressed as a set of keywords where each keyword is assigned a definite weight and each query is assigned the topic ids to identify each query in later stage.

2.4 Scoring

After construction of queries, each bag of words corresponding to each tweet is assigned a score with respect to a query. We have used two different scoring techniques for two separate runs.

Method–1: Co-occurrence based Similarity

This method is based on co-occurrence based similarity measure [2]. This method tries to find out how many words from the query have also occurred in the tweet and scored the tweet based on that. For a given tweet T = {t 1, t2 , ..., t n } and a given query Q = {q1, q2 , ..., qn } the score of the tweet is calculated as follows:

Score (T , Q) = | intersection of T ,Q | / | Q | , where | Q | denotes number of elements in set Q.

That is this score measure the fractions of common words in a tweet and a query. The higher the fraction, higher the probability that the tweet is relevant to the query.

Method–2: WordNet based Semantic Similarity

The previous method is generally based on the co-occurrence similarity which does not concern about the meaning wise similarity of two words. This problem can be solved by WordNet[3] based approach. WordNet is a lexical database of English. Each word in WordNet has a set of cognitive synonyms called synsets. Two find the similarity between two words we can calculate the similarity between two synsets.

For a given tweet T = {t1, t2 , ..., tn } and a given query Q = {q1, q2 , ..., qn } the score of the tweet is calculated as follows:

1) For each t i and q j we have first found the synsets of two words say S1 and S2 respectively. Now for each term in S1 and each term in S2 we have calculated wup similarity.[1] After this all wup score is added up and normalized. This normalized score denotes the similarity value between t i and q j
2) We iterate through all the terms in tweets and queries and summed up all the similarity score of each pairs and normalize it.
3) This normalize score is the final score of the tweet respect to that particular query.

2.5 Final Filtering

After scoring the tweets according to relevance to each topic, we need to choose most relevant tweets for a given topic. For this reason, we have taken a heuristically set threshold based filtering method to choose most relevant tweets. The threshold has been set to 0.25. That is the tweets which have a score greater than 0.25 are considered as relevant and are submitted. All other tweets have been discarded.

3 RESULT ANALYSIS

Table-1 shows the result of our two submitted runs. The run acquired from method-1 is tagged as "ss" and then run acquired from method-2 is tagged as "ws". The runs have been evaluated based ground truth obtained by the organizers. Different metrics like Precision@20, Recal@1000, MAP@10000 and MAP have been used to evaluate the runs.

Run Id	Precision @20	Recall @1000	MAP @1000	Overall MAP
trish_iiest_ss	0.0929	0.1407	0.0140	0.0203
trish_iiest_ws	0.0786	0.0618	0.0032	0.0099

Table-1

As it can be clearly seen from the result, though the second method uses a more deep similarity measure than the first approach the first approach performs better than the second one. The most probable reason for this is due to lack of grammar and spelling wise correctness of tweets. Most of the tweets are informally written microblogs, so using a standard English dictionary based filters and standard semantics based methods are not practically that much effective. While much simpler co-occurrence based similarity measure outperforms it on the basis of performance and running time and cost.

4 CONCLUSION

In this working note, we have presented a brief discussion on our approach to FIRE 2016 microblog task. We have observed that traditional dictionary and vocabulary based filtering techniques are very inefficient for informally written documents like tweets. The relatively simpler co-occurrence based methods suits well for future work that also includes finding new filtering techniques and parameters to tackle such informally written documents like tweets.

[1] http://search.cpan.org/dist/WordNet-Similarity/lib/WordNet/Similarity/wup.pm

5 REFERENCES

[1] http://nlp.stanford.edu/software/tagger.shtml.

[2] Ekkachai Naenudorn Suphakit Niwattanakul, Jatsada Singthongchai and Supachanun Wanapu. Using of jaccard coefficient for keywords similarity. volume 1, pages 380–384, Hong Kong, 2013

[3] https://**wordnet**.princeton.edu/

[4] S. Ghosh and K. Ghosh. Overview of the FIRE 2016 Microblog track: Information Extraction from Microblogs Posted during Disasters. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings. CEUR-WS.org, 2016.