TREC Microblog 2012 Track: Real-Time Algorithm for Microblog Ranking Systems

Davide Feltoni Gurini
Department of Engineering
Artificial Intelligence Laboratory
Roma Tre University
Via della Vasca Navale, 79 - 00146 Rome, Italy
feltoni@dia.uniroma3.it

Fabio Gasparetti
Department of Engineering
Artificial Intelligence Laboratory
Roma Tre University
Via della Vasca Navale, 79 - 00146 Rome, Italy
gaspare@dia.uniroma3.it

ABSTRACT

As a matter of fact Twitter is becoming the new big data container, due to the deep increase of amount of users and its growing popularity. Moreover the huge amount of user profiles and rough text data, are providing continuosly new research challenges.
This paper reports our contribution and results to the Trec 2012 Microblog Track. In this particular, challenge each participant is required to conduct a "real-time" retrieval task, which given a query topic seeks for the most recent and relevant tweets. We devised an effective real time ranking algorithm, avoiding heavy computational requirements. Our contribution is multifold: (1) adapting an existing ranking method BM25 to the microblogging purpose (2) enhancing traditional content-based features with knowledge extracted from Wikipedia, (3) employing Pseudo Relevance Feedback techniques for query expansion (4) performing text analysis such as ad-hoc text normalization and POS Tagging to limit noise data and better represent useful information.

1. INTRODUCTION

In the last year the huge interest in the microblog platform Twitter has been leading to large amounts of short text messages sent between users everyday. These short messages called "'tweets"' provide information ranging from job opportunities, personal opinions, newspaper articles to simply author's sentiments. The research community is now focused on different aspects such as limiting irrilevant or noisy information, employing ad-hoc POS Taggers, Named Entity Recognition tools or employing predictive analysis. In June 2012 the traffic data reached a rate of 400 million tweet per day. Consequently in order to explore the behavior and relevant features in the microblog sphere is thus required a lot of work to apply different linguist analysis techniques and data analysis methods. Our interest in microblog and social networks allowed us to partecipate in Trec Microblog 2012 Track based on Twitter corpus. Twitter corpus cover 2 weeks from Jan 23rd to Feb. 8th 2011 where TREC identified 60 query topics.

2. MICROBLOG AD-HOC TASK

2.1 Task

This year is the second time TREC conference releases a task in microblog track. The first challenge consists in finding and ranking relevant tweets given a query topic. This task is aimed to simulate user behaviours that want to retrieve some information on Twitter using few keywords. Basically, the goal is to retrieve the best relevant tweet pertinent to the given keyword and discard the irrilevant ones. In the evaluation, all retrieved document are ordered by recency and then evaluated by traditional metrics, such as Precision @30, R-precision, MAP and ROC curve. The judgment is based on 3-level scale: non-relevant, relevant and high-relevant. In Trec 2012, only high relevant tweets are taken into consideration.

2.2 Dataset

The corpus available for testing correspond to the one used in 2011, which consist approximately in 16 million documents gathered in the period 24-1-2011 and 8-02-2011. However we were able to collect around 15 million tweets due to protected users or tweet deletion. Every document contains many information for research such as information about the tweet and the user profile. We collected those tweets by means of web crawler and extract several features from the raw html source such as the url, number of retweets, if the tweet is a reply part of conversation, the location and so on. At the end of retrieval phase, two English language recognizers have been employed: TextCat[1] and Apache TIKA[2]. All the retweeted data are also discarded according with the TREC Microblog guidelines. Finally, the final corpus consists of 5 million tweets.

3. THE RETRIEVAL SYSTEM

Our approach is based on the idea that actual ranking algorithm based on *tf-idf* weighting schema and keyword matching can be a baseline for a microblog ranking system. Nevertheless this approach has large chances to be improved considering the peculiar characteristics of the domain.
We started building our system entirely using the open source

[1]http://textcat.sourceforge.net/
[2]http://tika.apache.org/

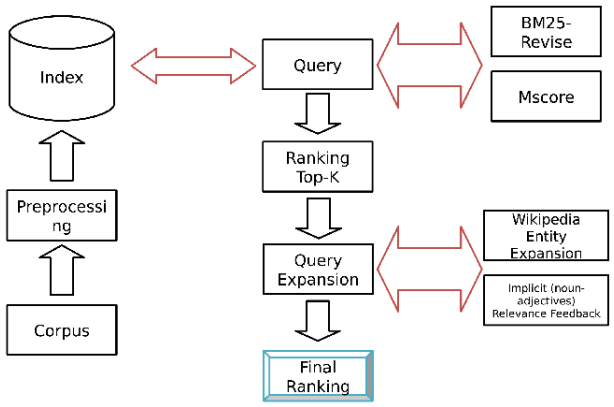

Figure 1: The Proposed Retrieval and Ranking System

information retrieval system Apache Lucene to store and index documents. As can be seen in the Figure 1 the first processing step of our approach is a ranking system based on BM25 formula implemented into Java Lucene[3] and a Microblog Scoring that will be explain in the next section. This first step is also used to retrieve the top-k relevant tweets and proceed with the query expansion process in order to improve accuracy of the ranking system.

3.1 Revised-BM25

Our BM25 algorithm approach is a variant of the standard BM25 ranking function. The differences between our function and standard BM25 are in the computation of term weights and the normalization of length weight. In both of the calculations we use the inverse function instead of the normal weight. Thus given a query q composed by query terms $\{q_1..q_n\}$, the score assigned to the document D is obtained:

$$BM25(q,D) = \sum_{i=1}^{n} IDF_{rt}(q_i) \cdot \frac{\frac{1}{tf(q_i,D)} \cdot (k+1)}{\frac{1}{tf(q_i,D)} \cdot (1-b+(b \cdot (LEN))} \quad (1)$$

where k and b are weight parameters set to $k = 2$ and $b = 0.67$[4], $avgl$(collection) is the average document lenght in the collection. LEN is the length weight normalization adjusted for short text and consequently for Twitter. We are assigning an higher score to longer tweets:

$$LEN = \frac{\frac{1}{LENGTH(D)}}{avgl(collection)}) \quad (2)$$

TF is the *term-frequency* and IDF is *inverted document frequency* calculated as follows:

$$IDF(q_i) = log \frac{N - docFreq + 0.5}{docFreq + 0.5} \quad (3)$$

[3]The first version of our Java implementation of BM25 ranking algorithm for Lucene, is available for download at http://code.google.com/p/bm25-lucene-score/ .
[4]This values are obtained by preliminary tests

	non relevant	relevant	high relevant
avg length (token)	8.58	10.74	10.21
avg # of retweet	8.91	0.34	0.41
avg url presence	52%	76%	94%
avg # of reply	0.12	0.04	0.02
avg # of hashtag	0.25	0.19	0.21
% of query terms	15%	40%	43%
avg noise terms	8%	3%	2%

Table 1: General statistic of Twitter features

where *docFreq* is the document frequency of the query word. Although Lucene provides high perfomance for indexing and searching, was hard to adapt it to the new algorithm BM25 because document frequencies and further implied features are deeply encoded in the Lucene system. Further algorithms that analyze specific social features extracted from the tweets and combined with the revised-BM25 ranking function are to be discussed in the next sections.

3.2 Micro-Score system

As a feature of microblogs, the limitation of text length forces users to write concisely. Sometimes this means that tweets do not directly include relevant content, but contain references or other microblog specific features to represent it. Table 1 summaries statistics of some relevant features in tweets judged non relevant, relevant or high relevant in 2011 Twitter Corpus.

It is clear that the average noise terms (of which we will discuss later) is inversely proportional to the relevance of the tweet while the urls occur more often in high relevant tweets. Based on that statistics we propose the generalized formula:

$$Mscore(T) = \alpha \cdot url + \beta \cdot log(rt) + \gamma \cdot reply + \delta \cdot log(noisetxt) \quad (4)$$

where the presence of some features are linearly combined altering the traditional ranking function. The first element ***url*** regards the presence of urls in a tweet. By using urls users can include many extra information such as an images, videos or correlated newspaper articles. Nonetheless we argue that analyzing every url and the page included in it, is not suitable for a real time ranking. Thus we decided to include just the feature that shows the url presence in the tweet.

Another microblog feature included in the formula is the number of ***retweets***. Preliminary experiments on small corpus used to assign weights to each feature show how this is not a so reliable indicator.

In the next step we classify the tweets in two class:

- conversational tweets
- direct tweets

Tweets are into the former category if they are a reply to another user tweet, or they are retweets. In this case we assign zero weight, 1 otherwise.

The last feature called ***noise terms*** or noise text, indicates tweets that contain *noisy information*. Our hypothesis is

that a relevant tweet contains well written text, e.g. syntactically correct.

Therefore we extracted a subset of tweets representing a noise text class:

Repeated Letter: Ex. looovee, soooo much

Alphanumeric Words: Ex. 2night, 4ever, str8

Strong presence of Smiley: Ex. `O_O`, `^_^`, :-), :)

In order to recognize those dirty text, we employed regular expression techniques. An example is given below:

Repeated Letter: `.*([a-zA-Z])\\1{2}.*`

Alphanumeric Words: `[0-9]{1}[a-zA-Z]+[a-zA-Z]+[0-9]{1}[a-zA-Z]*`

The outcome is a value close to 1 if the tweet contains an high level of syntactically incorrect content. The outcome is a value close to 0 if the tweet does not contains (in our hypothesis) noise text.
Afterwards that we calculated the Mscore formula placing a value for α , β , γ and δ parameter.
The linear combination between *Mscore* and *BM25-Revised* is our baseline algorithm for ranking tweet.

4. QUERY EXPANSION

Subsequently to a deep analysis of the microblog dataset, we discovered that some query terms are not able to retrieve all the relevant tweets. Basically, the recall is negatively altered because of the mismatch between the query keywords and the terms in the tweets (e.g., see the well-known vocabulary problem[6]. In order to improve the retrieval recall we decided to set up a *full automatic* query expansion module. Starting from top-15 documents ranked by our system, we follow two query expansion steps:

1. Wikipedia Topic-Entity Expansion
2. Noun-Adjective Pseudo Relevance Feedback

The following sections describe in details the steps.

4.1 Wikipedia Topic-Entity Expansion

In this first step of query expansion we used a service called *Wikify Service*[5] included in the Wikipedia Miner project of the Waikato University. This service automatically detects the *topics* mentioned a the given document, and also provides the probability that detected topics are right. Hence, we extracted the topics with best likelihood from **top-15** ranked tweets and built a vector representation for the semantic analysis.

Afterward we analyze the *semantic relatedness* between the extracted topics and the **original query terms** using the *Compare Service*[6] of Wikipedia Miner. Figure 2 shows an example of how semantic relation works in Wikimedia Miner. In few words the relatedness measures are calculated from the links going into and out of each page. Links that are common to both pages are used as evidence that they are related, while links that are unique to one or the other indicate the opposite. The page is the Wikipedia Page representing the term (if available). For the query expansion we suppose that if the extracted Wikipedia topic and the query term have an high semantic similarity, the topic can be quite good for the query expansion. The principal benefit of this approach is that, given ambiguous terms, such as NSA, we are able to obtain the potential meanings (e.g, National Security Agency, Information Security).

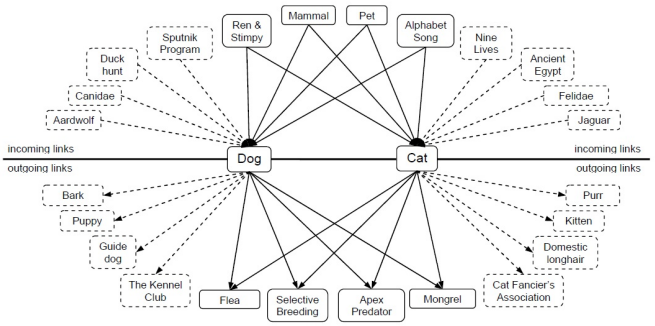

Figure 2: Wikipedia relation between cat and dog

4.2 Noun-Adjective Pseudo Relevance Feedback

The second step of query expansion analyzes the lexicon contained in top-15 ranked tweet. In order to automatically extract the terms for expansion we used the Pseudo Relevance Feedback approach. Basically we selected the top-15 ranked documents and, using a **Part of Speech Tagger**[7], we extracted the Proper Nouns (NNP) and Adjectives found in the documents. Finally, we ranked those terms with **tf-idf** weight schema, to better represent relevant terms. The entire process can be summarized as follows:

1. Select top-15 documents from First Retrieval
2. Process all documents with POS Tagger
3. Extract Noun and Adjectives words
4. Weight extracted words with ft-idf formula
5. Select top-3 extracted words for query expansion

At the end of Wikipedia and Pseudo Relevance Feedback processes, we combined both modules inserting the terms into the original query. If the terms - for some reasons - are not available, the following step is skipped. In order to assign a weigth for the *expanded terms* we used the Lucene boosting feature. Preliminary tests have led us to set to **1** the weight of the original query terms and **0.3** for the expanded terms.
An example of expansion is:

- <QUERY TOPIC>: 108
- <QUERY>: <identity theft protection>

[5]Live demo is available here: http://wikipedia-miner.cms.waikato.ac.nz/services/?wikify
[6]Live demo is available here: http://wikipedia-miner.cms.waikato.ac.nz/services/?compare
[7]We used the Stanford University Pos Tagger

- <EXPANDED>: <identity theft protection Crime^0.3 Fraud^0.3 Service^0.3 >

where all the query terms are combined by the OR operator. Thus, a possible query term vector is *identity theft fraud* or *identity protection service*.
Finally, the final *query expanded* is given in input to the Revised-BM25 and Mscore retrieval system, in order to rerank the documents.

Summary Statistics	
Run ID:	AIrun1
Task :	adhoc
Run type:	automatic
Collection crawl dates:	december 2011
Number of 200/301 tweets in crawl:	15000000
Collection crawl indexed:	HTML
Follows realtime constraints?	yes
Uses documents linked from tweets?	no
Uses other external resources?	yes
Number of Topics:	59
Total number of documents over all topics	
Retrieved:	58982
Relevant:	2572
Relevant retrieved:	1445
Mean average precision:	0.1522
R-precision:	0.1930

Figure 3: Summary of AIRUN1

5. EVALUATION

In this section we present the offical 2012 Trec evaluation for ad-hoc task.
We submitted only one run to microblog challenge. Our run -"*AIRUN1*"- uses no future evidence but employs external evidence such as Wikipedia Miner. Furthermore, we submitted a full automatic run without manual interation.
Figure 3 shows some statistics about our run such as MAP and R-Precision. Table 2 shows high relevant official results detected by TREC, and our self-made evaluation for all relevant results[8].
Figure 4 shows the difference between our median P@30 and median P@30 of all other runs, for each query topic. AIRUN1 outperforms the median in most of query-topics. However in few cases our system is quite under the median and our expectations. In detail we recognize the topics 81, 85, 89 in which our system has scored almost zero for P@30 and MAP. The query expansion activity process was not able to extract relevant adjectives and noun in the lexicon. Nevertheless our system is able to improve Lucene baseline score, in particular with query expansion and the MScore modules. In Table 3 we summarized the score of each module, from Lucene baseline to the full working system with BM25+Mscore+QueryExpansion modules. We achieved an improvement from 0.19 to 0.29 in Precision@5 using the combination of all the developed modules.

[8]The evaluation was performed using the same metrics released by the TREC

	All Relevant			**High Relevant**		
	P@5	P@30	MAP	R-prec	P@30	MAP
2012 AIRun	**0.501**	**0.40**	**0.331**	**0.193**	**0.2**	**0.152**

Table 2: 2012 High Relevant Official Results and All Relevant Unofficial Results

AIRUN1	P@5	P@30	MAP
Lucene baseline	0.19	0.12	0.08
+ Bm25(Rev.)	0.23	0.14	0.10
+ Mscore	0.27	0.17	0.13
+ Q.E.	0.29	0.208	0.15

Table 3: High relevant score evaluated for every module of our approach

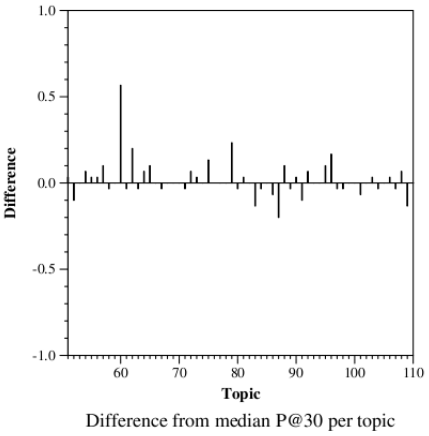

Figure 4: Difference -for each query topic- between median P@30 and AIRUN P@30

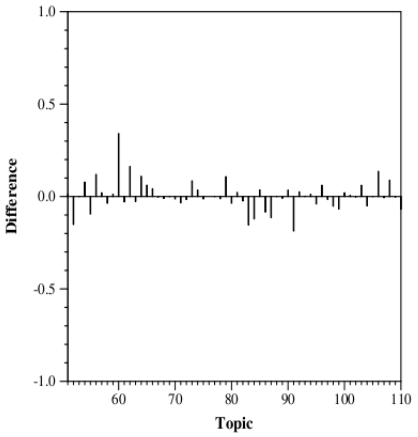

Figure 5: Difference from median average precision per topic

GROUP ID	P@30	GROUP ID	P@30
HIT MTLAB	0.2701	udel	0.196
ICTNET	0.2384	waterloo	0.1955
KobeU	0.2384	NC SILS	0.1938
PKUICST	0.2333	FUB	0.1932
CMU Callan	0.2333	QCRI	0.1921
ot	0.2328	UGENT IBCN SIS	0.1904
FASILKOMUI	0.2294	qcri twitsear	0.1898
IBM	0.2254	GUCAS	0.1876
uogTr	0.2232	SCIAITeam	0.1808
uiucGSLIS	0.2186	UvA	0.1774
PKUICST	0.2164	BAU	0.174
UWaterlooMDS	0.2107	udel fang	0.1616
york	0.2102	csiro	0.1616
XMU PANCHAO	0.2023	uog tw	0.1582
BUPT WILDCAT	0.2028	IIEIR	0.1508
AI ROMA3	0.1994	UEdinburgh	0.1226
IRIT	0.1983		

Table 4: List of groupID in the Ad-hoc task sorted by best P@30

6. CONCLUSIONS

In this paper we described the approach for the TREC 2012 Microblog Track. We propose an effective and straighforward real-time algorithm for ranking tweets that is able to exploit traditional retrieval and semantic annotation tools. The evaluation confirms that the approach is able to rank high relevant tweets with 0.2 Precision@30 higher than the median of all approaches.
It is interesting to take in consideration users interests, their older tweets and friends in order to create a basic user profiling. We are planning also to try some effective improvement in our query expansion system and in Microblog-Score. In particular we are studying techniques for identifying authoritative and influential users given a specific topic. In our opinion this combined approach can improve the ranking system, limiting issues related to spam documents. Finally we will investigate the improvement of discovering the *dirty text class*, and propose a Machine Learning approach to identify and automatically normalize this class of problems.

7. REFERENCES

[1] Apache Lucene. `https://apache.lucene.com`.

[2] Trec Microblog Track. `https://sites.google.com/site/microblogtrack/`, 2012.

[3] X. H. Bingqing Wang. Microblog track 2011 of fdu. 2011.

[4] C. Z. Charu C. Aggarval. *Mining Text Data*. 2012.

[5] D. Feltoni. Twittersa: un sistema per l'analisi del sentimento nelle reti sociali. Master's thesis, Roma Tre University, "2012".

[6] F. George and et al. The vocabulary problem in human-system communication. *Communications of the ACM*, 1987.

[7] D. R. Jaime Teevan and M. R. Morris. Twittersearch: a comparison of microblog search and web search. *Proceedings of the fourth ACM international conference on Web search*, 2011.

[8] M. Kaufmann. Syntactic normalization of twitter messages. 2010.

[9] R. Li and et al. Author model and negative feedback methods on trec 2011 microblog track. 2011.

[10] J. Prez-Iglesias. Integrating the probabilistic models bm25/bm25f into lucene. *Proceedings of the fourth ACM international conference on Web search*, 2009.

[11] S. Ravikumar and et al. Ranking tweets considering trust and relevance. *ACM 2012*, 2012.

[12] A. T. Rinkesh Nagmoti and M. D. Cock. Ranking approaches for microblog search. *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2010.

[13] I. Soboroff and E. M. Voorhees. Overview of the trec 2011 web track. *Proceedings of Trec 2011*, 2011.