Effectiveness of State-of-the-art Features for Microblog Search

Firas Damak
IRIT UMR 5505 CNRS, France
University of Toulouse
damak@irit.fr

Karen Pinel-Sauvagnat
IRIT UMR 5505 CNRS, France
University of Toulouse
sauvagnat@irit.fr

Guillaume Cabanac
IRIT UMR 5505 CNRS, France
University of Toulouse
cabanac@irit.fr

Mohand Boughanem
IRIT UMR 5505 CNRS, France
University of Toulouse
boughanem@irit.fr

ABSTRACT

We investigate in this paper information retrieval in microblogs exploiting different state-of-the-art features. Microbloggers, besides posting microblogs, search for fresh and relevant information related to their interests, by submitting a query to a microblog search engine. The majority of approaches that collect information from microblogs exploit features such as the recency of the microblog, the authority of his/her author... to improve the quality of their results. In this paper, we evaluated some of the state-of-the-art features to determine those that discriminate relevant from irrelevant microblogs given an information need. Then, we used the selected features to learn models to determine their effectiveness in a microblog search task. We conducted a series of experiments using the dataset and topics of the TREC Microblog 2011 and 2012 tracks. Results show that content, hypertextuality, and recency are the best predictors of relevance. We also found that Naive Bayes was the most effective learning approach for this type of classification.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval—*Retrieval models*

Keywords

Information retrieval, Microblog search, Feature evaluation

1. INTRODUCTION

Searching for real time information published in microblogging platforms has become a challenge for Information Retrieval (IR) systems. Since a huge number of microblogs are published and many queries are issued each day (200 million tweets[1] and over 1.6 million search queries[2] in Twitter), this new type of media has become of high interest for information retrieval research. IR systems should be adapted to the specificities of microblogs and be effective to handle the relative exponential growth of information requests.

Microblogs have many specificities. On the one hand, people's motivations for microblog search are diverse [10]: some are similar to those of web search (e.g., `astronomy or science stuff`), while others, regarding a need for timely (e.g., `traffic jam`, `down services`) or social information (e.g., `people with similar interests`), are relatively specific to microblog search. On the other hand, microblogs are different from ordinary web pages. Tweets (i.e., microblogs in Twitter), for example, cannot exceed 140 characters. They are usually composed of a single sentence. Users publish microblogs on a wide variety of topics in real time. Thus, the messages contain mistakes and some are written in netspeak. Besides, some microblogging platforms like Twitter use specific syntax (e.g., re-tweets, hashtags, mentions), that is likely to challenge conventional natural language processing techniques. Finally microblogs often point to external sources by including hyperlinks to documents or other contents on the web.

Having in mind all the aforementioned specificities, the literature mentions tailored IR approaches [8]. In addition to message content, relevance is evaluated by combining other factors to process these specificities. It may depend on the novelty of the information, the quality of the message, the author's authority, and the freshness of the message. These factors are taken into account by combining specific features into the relevance function used. For example, if we consider the author's authority factor in Twitter, the associated features could be the number of the tweets of the author and the number of his followers [7]. One may also consider the number of times a user has been referred to or the rank of the author according to a PageRank-like algorithm based on retweet relations [2].

While the effectiveness of each system can be assessed as a whole, we failed to find any work probing the effectiveness of each underlying feature in the literature. In this paper,

SAC'13 March 18-22, 2013, Coimbra, Portugal.

[1] http://blog.twitter.com/2011/06/200-million-tweets-per-day.html
[2] http://engineering.twitter.com/2011/05/engineering-behind-twitters-new-search.html

we aim at evaluating the features commonly exploited in microblog search approaches. More specifically, we attempt to select those most *effective* for microblogs search.

The paper is organized as follows. In Section 2 we discuss related works for microblog search using features. Our evaluation method is then presented in Section 3. Section 4 presents our results on two TREC test sets that are then discussed in Section 5.

2. RELATED WORK

Many approaches using features to extract information from microblogs have recently been introduced in the literature. Many of them have been evaluated using the TREC Microblog track.[3] The Microblog track, started in 2011, promotes search tasks and evaluation methodologies for information seeking behaviors in microblogging environments.

Among these approaches Duan et al. [2] used features to search and to rank microblogs given a query. These authors relied on the Okapi BM25 model as a content relevance feature. Some features specific to Twitter (i.e., popularity of the tweet, retweet frequency, hashtags frequency, length of the tweet, URL presence, is-reply) and others were used to measure the importance of the authors (i.e., number of followers, number of mentions). A machine learning approach was then used to build the search model. Similarly, the approach described in [6], 1st ranked in TREC 2011 Microblog track, used a learning approach to rank tweets based on a set of features (i.e., text score, time difference score, hashtag presence, URL presence, length, is-reply and the percentage of terms in a tweet that are out of the English vocabulary).

Features have also been used as quality indicators of microblogs. To select the best message from a set of retweets, Massoudi et al. [5] defined various quality indicators, such as the presence of a URL, the number of followers of the user who rebroadcast the microblog, and the time elapsed before its republication. With the same purpose, authors in [11] used a machine learning model based on content-based features (i.e., punctuation and spelling features, syntactic and semantic complexity and grammaticality), link-based features (i.e., URL presence, is-retweet, number of mentions, hashtag presence, number of times a tweet has been retweeted, URL reputation, hashtag reputation) and temporal features (i.e., the recency of the tweet).

According to [8], apart from content, features may be classified (Table 1) according to two characteristics: External features use information outside the microblog dataset (e.g., Wikipedia or the Web). Future features use information that would not have been available to the system when the query was issued.

In this paper, we are only interested in realistic settings: we do not want to use any information that has been published after the timestamp of the queries. The features under study were scored using only the available information in the corpus used for this study (e.g., number of tweets for a user will not concern the number of his/her tweets in all Twitter, but only in the corpus used and before the timestamp of the query; or, the number of followers was not considered since we did not have required information in the corpus). Approaches using features marked as future in Table 1 are not necessary unrealistic: some approaches do not take into account the timestamps of queries. They thus relied on the online version of Twitter. However, in our case, the same features are considered as unrealistic since we consider the query timestamps. The complete list of selected features we study in this paper are presented in the next section. We evaluated then the selected features to determine those that best reflect relevance. Finally, we used the relevant features to learn models to determine their effectiveness in a microblog search task.

Table 1: External (E) and future (F) features

Feature		E	F
Popularity of the tweet	[2]	-	-
Topic tweet term matching	[1]	-	-
Retweet frequency	[14, 4, 11, 2]	-	+
Hashtags frequency	[2]	-	-
Hashtags presence	[11, 6]	-	-
Hashtag reputation	[11]	-	+
Length of the tweet	[14, 4, 6, 2]	-	-
Url presence	[11, 5, 6, 2]	-	-
Number of urls in the tweet	[14]	-	-
Url reputation	[11]	-	+
Is-reply	[11, 6, 2]	-	-
Number of tweets of a user	[14]	-	+
Number of followers	[4, 5, 2, 14]	-	+
Number of mention	[11, 2]	-	+
Tweet time difference	[4, 11, 6]	-	-
Language quality	[6]	+	-

3. METHOD

The corpus we used for our study was provided by the TREC evaluation campaign (TREC Microblog Track) and is based on Twitter. We describe it in section 3.2. Section 3.1 describes the features selected for analysis.

3.1 Selected Features

Most of the evaluated features are derived from state of the art. We evaluate these features in a context of information retrieval, i.e., the evaluation of the relevance of a tweet given a query. We use the following notations in the sequel: t is a tweet posted a timestamp $tmp(t)$, q is the query, also known as a topic, which has a timestamp $tmp(q)$, C_q is the corpus of tweets posted before the timestamp of the topic q. T_q is the set of tweets returned by a given search engine based only on the content of tweets. $T_q \subseteq C_q$.

The features are normalized to lie between 0 and 1.

Tweet popularity [2]: estimates the popularity of tweet t in T_q. The similarity between a pair of tweets is calculated using the Lucene similarity function[4] $sim(\vec{t}, \vec{t'})$. We denote the current tweet modelled as a vector $\vec{t}$:

$$F_1(\vec{t}, q) = \frac{\sum_{\vec{t'} \neq \vec{t}} sim(\vec{t}, \vec{t'})}{|T_q| - 1} \quad (1)$$

Tweet length [2]: instinctively, the longer a sentence is, the more information it is able to contain. We calculate this feature by counting the number of words in the tweet. We denote $l(t)$ the number of words in tweet t. This feature is defined as:

$$F_2(t, q) = \frac{l(t)}{max_{t' \in T_q} l(t')} \quad (2)$$

[3] http://sites.google.com/site/trecmicroblogtrack/

[4] http://lucene.apache.org/core/3_6_1/scoring.html

Exact term matching: promotes tweets that contain terms of the topic q. $nb(t,q)$ denotes the number of common terms between t and q:

$$F_3(t,q) = \frac{nb(t,q)}{max_{t' \in T_q} nb(t',q)} \quad (3)$$

URL presence [7]. By sharing an URL, an author would confirm the information published in his tweet:

$$F_4(t) = \begin{cases} 1 & \text{if } t \text{ contains at least one URL} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

URL frequency [14]: counts how many URLs are published in tweet t:

$$F_5(t,q) = \frac{|\{w \in t / isURL(w)\}|}{max_{t' \in T_q} |\{w \in t' / isURL(w)\}|} \quad (5)$$

URL popularity: aims at calculating how important the URLs published in tweet t are in C_q. $freq(url)$ denotes the number of time the URL appears in C_q:

$$F_6(t,q) = \frac{\sum_{url \in t} freq(url)}{max_{t' \in T_q} \sum_{url \in t'} freq(url)} \quad (6)$$

Notice that different URLs may lead to the same document. Such URLs are not uniformed in this work. they are considered as different ones.

Hashtag presence [6]. The # symbol, called a hashtag, is used to mark a keyword or a topic in a tweet. Any user can categorize or follow topics with hashtags.

$$F_7(t) = \begin{cases} 1 & \text{if } t \text{ contains a hashtag} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Hashtag popularity [2]: $freq(h)$ denotes the frequency of a hashtag h in the corpus C_q:

$$F_8(t,q) = \frac{\sum_{h \in t \wedge isHashtag(h)} freq(h)}{max_{t' \in T_q} \sum_{h \in t'} freq(h)} \quad (8)$$

Topic as hashtags: calculates the number of terms of topic q that are present as hashtag in tweet t:

$$F_9(t,q) = \frac{|\{w \in q / \#w' \in t\}|}{max_{t' \in T_q} |\{w \in q / \#w' \in t'\}|} \quad (9)$$

Number of tweets [7]. $N(a(t))$ is the number of tweets published by the author of the tweet t in C_q:

$$F_{10}(t,q) = \frac{N(a(t))}{max_{t' \in T_q} N(a(t'))} \quad (10)$$

Mention [14]. $M(a(t))$ denotes how many times the author of the tweet t has been mentioned in C_q:

$$F_{11}(t,q) = \frac{M(a(t))}{max_{t' \in T_q} M(a(t'))} \quad (11)$$

Is-reply [6]. If a user likes a tweet published by another user, he/she may comment and publish it again. In this case, the new message will be marked with *RT* (retweet).

$$F_{12}(t) = \begin{cases} 1 & \text{if } t \text{ contains RT} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

Recency [4]. The difference in time, measured in seconds, between the query q and the tweet t. $tmp(t)$ is the timestamp in seconds of the tweet t.

$$F_{13}(t,q) = \frac{tmp(q) - tmp(t)}{max_{t' \in T_q} tmp(q) - tmp(t')} \quad (13)$$

Language quality [2]: is used to approximate the language quality of tweets. It calculates the proportion of existing words in a dictionary[5] with respect to all the terms of the tweet t. $dic(term)$ return a binary value: 1 if the term exists in the dictionary, 0 if not:

$$F_{14}(t) = \frac{\sum_{term \in t} dic(term)}{l(t)} \quad (14)$$

3.2 Corpus

We used for our study the corpus of tweets provided to the 2011 and 2012 TREC Microblog Track participants. The collection is composed of 16 million tweets expressed in various languages and posted on Twitter between January 23, 2011 and February 8, 2011. Each tweet is characterized by its identifier (`ID`), author, and date of publication. Topics are composed of multiple tags. The `title` tag describes the user's information need expressed at a given time `query-time`: 49 topics of the 2011 track and 60 topics of the 2012 track. Relevance judgements were obtained from qrels of Microblog Track 2011 and 2012. The relevance of each tweet is ternary: irrelevant, relevant, and highly relevant. However, we did not differentiate between relevant and highly relevant in our evaluation.

Our study was conducted in two phases: we first evaluated the features using score distribution and some attribute selection approaches. We used in this case topics of TREC 2011. Different sets of features are obtained from this studies. Then, we evaluated the effectiveness of these features by applying the different outcome sets on learning techniques. We used for this second case the topics of TREC 2011 for learning and topics of TREC 2012 for the evaluation.

3.3 Preliminary Studies

We describe in this section the methodologies used to evaluate the effectiveness of the selected features. We conducted two different studies: one based on feature score distribution, and another using some attribute selection approaches.

3.3.1 Score Distribution Study

The assumption behind this study is that an effective feature discriminates the relevant tweets from irrelevant ones (i.e., effective features will not have the same distribution in relevant tweets compared to irrelevant tweets). To assess a feature, we looked at its score distribution within microblogs. We compared its scores between relevant and irrelevant microblogs. The set of tweets to analyse was formed as follows: for each topic having more than 100 relevant tweets[6], we kept all the relevant tweets according to the qrels, and we added the same number of irrelevant tweets. Irrelevant tweets were selected according to their *content score*, i.e., we only kept the *most relevant* according to a search engine. We used Lucene[7] as a search engine for this purpose. Resuls of all topics were merged together to plot scores for the overall distribution.

3.3.2 Attribute Selection Approaches Study

[5]http://code.google.com/p/language-detection/

[6]We only kept these topics to have a sufficient number of tweets to study. This corresponds to 9 topics (to be compared with the initial number of 49)

[7]http://lucene.apache.org/

In a second study, we relied on some selection attribute algorithms to determine the best features to use in a microblog search information task. Attribute selection algorithms [3] aim at identifying and removing as much of the irrelevant and redundant information as possible. We used Weka[8] for this experiment. It is a powerful open-source Java-based machine learning workbench that brings together many machine learning and selection attributes algorithm.

We proceeded thereby: the top 1,500 tweets for each topic were retrieved from Lucene. Then, all feature scores were calculated for each tweet. We identified the relevant and irrelevant tweets according to the qrels of the topics. The obtained tweet set contains 72,614 instances composed of 2,129 relevant tweets and 70,485 irrelevant tweets. One could observe that this collection has unbalanced relevance class distribution. This occurs when there are much more items in one class than in the other class of a training dataset. In this case, a classifier usually tends to predict samples from the majority class and completely ignore the minority class [12]. For this reason we applied an under-sampling approach (reduce the number of samples which have the majority class) to generate an unbalanced collection composed of 2,129 relevant and 2,129 irrelevant tweets that were randomly selected. Finally, we applied the attribute selection algorithm on the whole set.

4. RESULTS OF ATTRIBUTE SELECTION

In this section, we present results of preliminary studies. Then, we compare and discuss our findings.

4.1 Score Distribution Study

Figure 1 shows the distribution of the scores of relevant tweets (Relevant) and irrelevant tweets (Not relevant) for our first study. The intervals were calculated with the Sturge's rule [9]. One can observe that Tweet popularity (F1), Length (F2), Exact term matching (F3), URL presence (F4), URL frequency (F5), URL popularity (F6) and Recency (F13) are not distributed the same way compared to irrelevant tweets. These features have their best scores in relevant tweets and characterize more relevance. The difference of the two populations (relevant and irrelevant) is statistically significant according to Student's t test, bilateral and paired with $p < 0.05$.

4.2 Attribute Selection Approaches Study

Table 2 shows the features selected through selection attribute algorithms. Practically, features obtained from the score distribution study and most of the selection attribute approaches are the same (F1, F2, F3, F4, F5, F6, F13), which confirms the effectiveness of this set compared with the rest of features. We also conducted this analysis without sampling the dataset. We found that there is no effect of sampling on the outcome of attribute selection approaches.

4.3 Discussion

We found that the same features that emerge from the first study were selected by selection attribute approaches: content features (Tweet popularity, Length, Exact term matching), Hypertextuality features (URL presence, URL popularity, URL frequency) and time feature (Recency).

[8]http://www.cs.waikato.ac.nz/ml

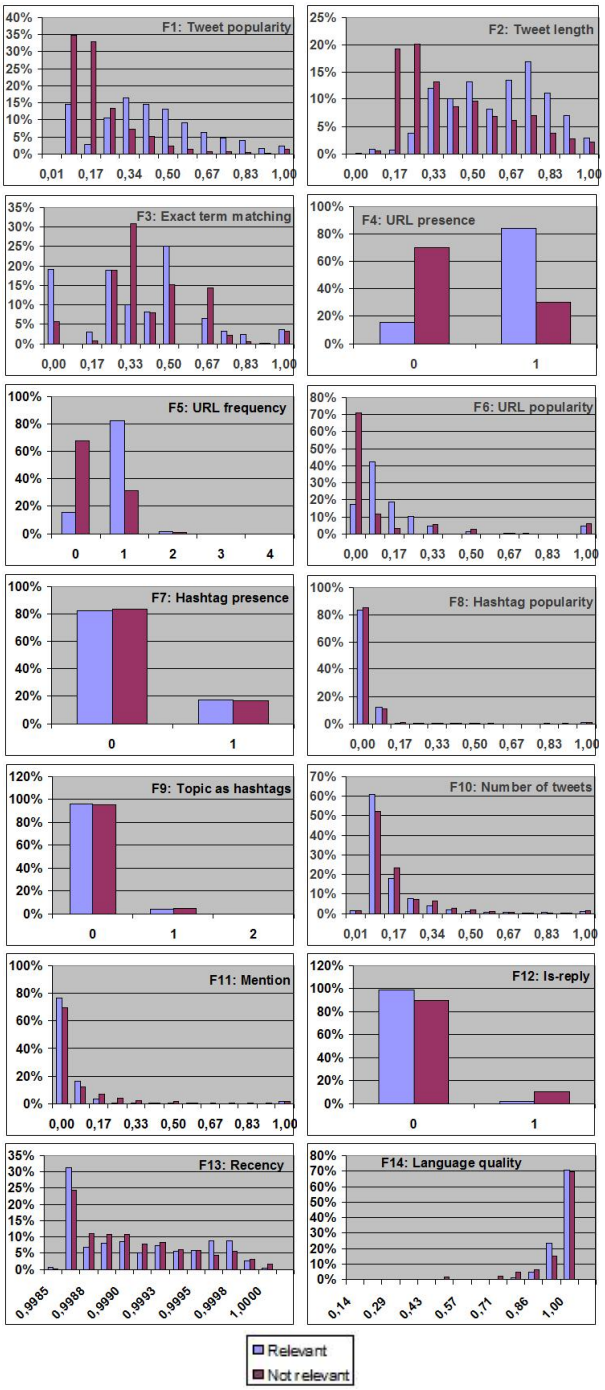

Figure 1: Distribution of the scores of relevant versus irrelevant tweets

Other features were selected by selection attribute algorithms, but less frequently: author authority features (Number of tweets, Mention, Topic as hashtags) and Language quality feature. Finally, the hashtag features (Hashtag popularity, Hashtag presence) were seldom selected and seem not to be effective for assessing relevance.

Table 2: Selected features with attribute selection approaches

Algorithm	Lucene	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
Cfssubseteval	+	+	+	+	+	+	+						+	+	
ChisquaredAtt.Eval	+	+	+	+	+	+	+			+	+	+	+	+	+
FilteredAtt.Eval	+	+	+	+	+	+	+			+	+	+	+	+	+
FilteredSubsetEval	+	+	+	+	+	+	+							+	
Gain ration att eval	+	+	+	+	+	+	+			+	+	+	+	+	+
Info gain att eval	+	+	+	+	+	+	+			+	+	+	+	+	+
One att eval	+	+	+	+	+	+	+			+	+	+	+	+	+
ReliefFAttribute Eval	+	+	+	+	+	+	+			+	+	+	+	+	+
SVM Attribute Eval	+	+	+	+	+	+	+			+		+	+	+	+
SymetricalUncertEval	+	+	+	+	+	+	+			+	+	+	+	+	+
Consistency subset Eval	+	+	+	+	+	+	+			+	+	+	+	+	+
Wrapper subset Eval	+			+	+	+	+								
LatentSymanticAnalysis	+	+	+	+											
Count	13	12	12	13	12	12	12	0	0	9	8	9	10	11	9

5. EFFECTIVENESS OF THE SELECTED FEATURES

In this section, we evaluate machine learning techniques with the set of effective features identified in the previous section. The aim is twofold: on the one hand we wondered whether the attribute selection really improves the results of a microblog search task. On the other hand, we intended to measure the performance of some learning algorithms in this type of classification.

5.1 Results

To evaluate the learned models, we used Lucene results of the 2011 TREC Microblog topics results as a learning set and the 2012 TREC Microblog topics results as a test set. Learned model predicts the relevance class (relevant or not relevant) for resulting tweets and give effectiveness classification scores. The non-relevant predicted tweets are then deleted and tweets predicted as relevant are ranked given the effectiveness classification scores. To evaluate our runs, we used the P@30 measure, since it was the official measure for the 2011 Microblog track of TREC.

We chose to experiment with three learning algorithms. This choice is explained by the fact that they are the most used in the short texts classification tasks and they have been shown to be effective for microblogs search in the past: SVM [11, 2], J48 [13] and Naive Bayes [13].

Authors in [3] studied the effectiveness of some selection attribute approaches with learning algorithms. Based on their study, we used the same combination of learning and attribute selection approaches applied on our own features:

- Naive Bayes and Wrapper Subset Evaluation (WRP) which uses a target learning algorithm to estimate the worth of attribute subsets. Thus, the features selected in this case were Lucene score, F3, F4, F5, F6.
- Naive Bayes and Correlation-based feature Selection (CFS) (Lucene, F1, F2, F3, F4, F5, F6, F12, F13).
- J48 and ReliefFAttribute Eval (RLF) (Lucene, F1, F2, F3, F5, F6, F9, F10, F11, F12, F13, F14).
- SVM and SVM Attribute Eval that assess attributes using the SVM classifier (Lucene, F1, F2, F3, F4, F5, F6, F9, F11, F12, F13, F14).

Table 3 shows results of the 3 learning approaches learned with score distribution outcome features, attribute selection approaches features, and with all features. Results were compared with the run called Lucene (baseline: only Lucene scores were used to rank tweets).

5.2 Discussion and limitations

Our fist aim for this study is to check if attribute selection improves effectiveness of learned models. SVM attribute Eval, WRP, CFS, and score distribution outcome features confirm the hypothesis. Apart from J48 learned with RLF selected feature, all results were improved compared to runs for which all features are used. One could also see also that apart from J48, machine learning approaches have better effectiveness with attribute selection approaches than when they are learned with our score distribution study features.

Our second aim was to find which machine learning approach is the more suitable for a microblog search task. We notice that only Naive Bayes model outperforms Lucene (+18% with CFS selected features and +16% with score distribution selected feature). The other learning approaches failed to improve effectiveness, leading us to conclude on the interest of Naive Bayes.

We compared the run obtained using Naive Bayes learned with CFS features with TREC 2011 Microblog participants. We learned and cross-validated Naive Bayes with CFS selected feature on TREC 2011 topics. We obtain an averaged P@30 of 0.3707. This result would have been ranked 5th among participants who did not use future evidences and submitted automatic runs. This precision is reduced by 9.4% when using the same model on TREC 2012 topics. In addition, machine learning approaches such as J48 and SVM have a gain of 80% of effectiveness when tested and cross-validated on TREC 2011 topics. However, they did not perform as intended on 2012 topics. All these observations raise the question whether the two track topics and qrels have been created in the same way.

In order to control this potential collection bias, we also merged 2011 and 2012 tracks topics and repeated the same steps. We obtained an averaged P@30 of 0.3435. This good result confirms that Naive Bayes learned with CFS selected features is the most suitable for this task.

Nonetheless, our work presents some limitations. First, machine learning certainly improves effectiveness of classifi-

Table 3: Results (P@30), results in bold indicate significant improvement over the baseline

Lucene (Baseline)	0.2842		
	Score distribution features	Attribute selection features	All features
J48	0.1627	0.0983 (RLF)	0.1000
Naive Bayes	**0.3305**	**0.3311** (WRP) **0.3356** (CFS)	0.2372
SVM	0.1689	0.1746 (SVM)	0.1729

cation. However, the weight of features is not always accessible, neither how features were combined. Secondly, some features which are used in some approaches of microblog search (e.g., retweet frequency, number of followers of an author) could not be evaluated since the required data do not appear in the corpus we used for experimenting. An open access to Twitter would be necessary to evaluate them, which is not possible.

Finally, concerning the TREC Microblog track, participating teams promoted microblogs published by popular, well-retweeted, with lots of mentions authors. However, it seems in reality that these features do not reflect relevance for our search task. Indeed, there are cases when people search for microblogs and find very relevant information from people who are not popular. Those people just happen to be in the right place and time to post the relevant and fresh information. One flaw of the track is that the tasks consist only in finding relevant microblogs given an information need. However, a microblogger could be interested in other entities like conversations, persons, trending topics .... The 2011 and 2012 tasks do not deal with such a broad range of search types.

6. CONCLUSION

In this paper, we analysed the effectiveness of a set of features commonly used in microblog search. We compared the distribution of scores between relevant and irrelevant microblogs for selected topics. We also used selection attribute approaches for the same purpose. We found almost the same results: tweet popularity, length, exact term matching, URL presence, URL popularity in the corpus, URL frequency in the microblog and the recency of the microblog are the features that best reflect relevance. We then tried to compare often-used learning approaches for microblog task and we found that only Naive bayes yields improvement in terms of effectiveness compared to our baseline. Regarding future work, we will try to improve recall by using query expansion and harnessing the content of URLs published in microblogs.

7. REFERENCES

[1] F. Damak, L. B. Jabeur, G. Cabanac, K. Pinel-Sauvagnat, L. Lechani, and M. Boughanem. IRIT at TREC Microblog 2011. In *TREC'11 : 20th Text Retrieval Conference*. NIST, Nov. 2011.

[2] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.-Y. Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 295–303, Stroudsburg, PA, USA, 2010.

[3] M. A. Hall and G. Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Trans. on Knowl. and Data Eng.*, 15(6):1437–1447, Nov. 2003.

[4] M. Magnani, D. Montesi, and L. Rossi. Conversation retrieval for microblogging sites. *Inf. Retr.*, 15(3-4):354–372, 2012.

[5] K. Massoudi, M. Tsagkias, M. de Rijke, and W. Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 362–367, Berlin, Heidelberg, 2011. Springer-Verlag.

[6] D. Metzler and C. Cai. USC/ISI at TREC 2011: Microblog Track. In *TREC'11 : 20th Text Retrieval Conference*. NIST, Nov. 2011.

[7] R. Nagmoti, A. Teredesai, and M. De Cock. Ranking approaches for microblog search. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, pages 153–157, Washington, DC, USA, 2010.

[8] I. Ounis, J. Lin, and I. Soboroff. Overview of the TREC-2011 Microblog Track. In *TREC'11 : 20th Text Retrieval Conference*. NIST, 2011.

[9] H. A. Sturges. The Choice of a Class Interval. *Journal of the American Statistical Association*, 21(153):65–66, Mar. 1926.

[10] J. Teevan, D. Ramage, and M. R. Morris. #twittersearch: a comparison of microblog search and web search. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 35–44, New York, NY, USA, 2011. ACM.

[11] J. Vosecky, K. W.-T. Leung, and W. Ng. Searching for quality microblog posts: Filtering and ranking based on content analysis and implicit links. In *DASFAA (1)*, pages 397–413, 2012.

[12] S.-J. Yen and Y.-S. Lee. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In *Intelligent Control and Automation*, volume 344 of *Lecture Notes in Control and Information Sciences*, pages 731–740. Springer Berlin / Heidelberg, 2006.

[13] Q. Yuan, G. Cong, and N. M. Thalmann. Enhancing naive bayes with various smoothing methods for short text classification. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 645–646, New York, NY, USA, 2012. ACM.

[14] L. Zhao, Y. Zeng, and N. Zhong. A weighted multi-factor algorithm for microblog search. In *Proceedings of the 7th international conference on Active media technology*, AMT'11, pages 153–161, Berlin, Heidelberg, 2011. Springer-Verlag.