FUB, IASI-CNR, UNIVAQ at TREC 2011 Microblog track

Gianni Amati[1], Giuseppe Amodeo[1,2,3], Marco Bianchi[1], Alessandro Celi[3], Cesidio Di Nicola[3], Michele Flammini[3], Carlo Gaibisso[2], Giorgio Gambosi[2], and Giuseppe Marcone[1]

[1] Fondazione Ugo Bordoni
gba,gamodeo,mbianchi,gmarcone@fub.it
[2] IASI-CNR
gaibisso@iasi.cnr.it, gambosi@mat.uniroma2.it
[3] Università dell'Aquila
celi,flammini@univaq.it

1 Introduction

The ad-hoc task of the microblogging track has an important theoretical impact for Information Retrieval. A key problem in Information Retrieval is, in fact, how to compare term frequencies among documents of different length. Apparently, term frequency normalization for microblogging can be simplified because of the short length constraint for the composition of admissible messages. The shortness of messages reduces the number of admissible values for the document length, and thus the length of a message can be regarded as if it were almost small and constant. On the other hand, short messages can carry a small amount of information, so that they are hardly distinguishable from each other for content. To overcome both problems, we propose to use a precise mathematical definition of information as the one given by Shannon [10] to provide an ad hoc IR model for Microblogging search. We show how to use Shannon's information theory and coding theory to weight the query content in Twitter messages and retrieve relevant messages.

A second major problem of the microblogging track, as well as of any new collection of TREC, is the unavailability of a set of queries to derive and tune model parameters. Moreover, this is the first evaluation campaign on a new released corpus ever made for the microblogging search task, and in absence of any relevance data, it seems very interesting to define a retrieval methodology which is independent from relevance data, but still highly effective for the ranking of very short messages. Indeed, the proposed information theoretic methodology leads to the construction of a microblogging retrieval model that does not contain parameters to learn, and evaluation has shown the effectiveness of such parameter-free approach.

In addition to these two major problems (i.e. how short length affects relevance and how to learn parameters in absence of any relevance data), message recency is the only criterion applied to re-rank the retrieved messages. Thus, we regard the microblogging task more as a filtering decision task than as a ranking task.

The new microblogging search task shares several similarities with some of the previous TREC evaluation campaigns, in particular with the past legal and blog TREC tracks. The legal track is basically a filtering task, that provides a large boolean retrieved set. In the legal track of TREC 2008 [8], for example, participants were asked to improve the quality of a given boolean baseline. This baseline was hard to improve according to the official evaluation measure. The task was to perform a dynamic cut-off value K in the ranking, being all evaluation measures estimated at the depth of this variable value for K, e.g. the precision and recall at depth K, $P@K$ or $R@K$ or other similar official measures used to assess the quality of the retrieved set.

Similarly to the recency re-ordering of retrieved messages, in the TREC 2008 blog track [6] participants had to re-rank the documents by relevance according also to an opinion content dimension. An evaluation study however showed that filtering relevant documents by a second dimension or criterion, such as the

opinionated content of documents, is often more harmful than performing a mild re-ranking strategy for the official MAP or P@10 measures [3] or even no re-ordering at all.

As a consequence of these general remarks we made the following hypotheses and submissions:

a) We have submitted a standard TREC baseline (the run named DFReeKLIM with 1000 messages retrieved per query) ordered by relevance only, that is without any time analysis, in order to assess how time re-ranking affects the precision at different depths of the retrieved set.
b) Relevance by score distribution follows a mixture of two distributions (e.g. one exponential for the non-relevant documents, and the normal for the relevant ones [7,11]). Therefore, relevant documents have to be found on top of the ranking, and indeed $P@K$ is a decreasing function with respect to K. It has been also observed that the precision at depth K, $P@K$, increases with collection size [5]. We assumed that relevance is dependent of recency of the messages. However, a pure re-ranking by time of the first K topmost messages, retrieved by relevance, hardly improve the official measure $P@30$ when $K \neq 30$ since relevance scores and relevance ranks do not distribute uniformly but follow a power law. Therefore we submitted an official run retrieving exactly 30 messages per query (the run named DFReeKLIM30).
c) We made a preliminary recency analysis. A dynamical cut to the retrieved set was introduced. The aim was to predict the best K documents for each query for which time reordering would have been successful. The mean threshold value for K was 73. The effectiveness of the methodology (run DFReeKLIMDC) must be assessed by the evaluation measures on time re-ranking.
d) We explicitly assumed dependence of relevance with respect to time and used the time ranking as recency score to reorder by relevance the first pass retrieval. The effectiveness of the methodology can be thus assessed by the evaluation measures on relevance and not by time re-ranking, as performed by the official TREC measures (run DFReeKLIMRA).

2 The Tweets2011 benchmark and experimentation settings

The Microblog track of TREC 2011 is based on the Tweets2011 corpus, a collection made up of messages sampled from the Twitter public timeline over a period of 2 weeks across January and February 2011. The Tweets2011 corpus contains approximately 16 million tweets, including replies, retweets and spam tweets. The corpus must be downloaded directly from Twitter from an official list of tweets and by means of a software tool provided by the track organizers, and building either a JSON or a HTML format collection. In the first case the tool would have taken a prohibitive time since it invokes the public Twitter API whose usage is limited to 150 API calls per hour. Because of this restriction *we had to build a HMTL format collection*. In this case the tool reconstructed the original tweets crawling the Twitter Web site without any rate limit. Unfortunately the HTML format is not as rich as the JSON format.

We parsed the constructed HTML collection to obtain a standard TREC format collection. Furthermore, we filtered re-tweets (i.e. tweets having the `RT:` prefix), and we tagged in the `TEXT` fields all tiny urls, labels (identified by the `#` character), replies and mentions (both identified by the `@` character). At the end of the process each tweet had the following format:

```
<DOC>
<DOCNO>2896857361547264O</DOCNO>
<DATE>Sun Jan 23 00:13:40 +0000 2011</DATE>
<TEXT>
      <LA>#Twitition</LA> this is a answer of a hater to me ):
      <A>http://twitition.com/p6q53</A> <TO>@TwitterUserNameToReply</TO>
</TEXT>
```

```
<SCREENNAME>TwitterUserName</SCREENNAME>
<FULLNAME>Mrs. XXXX </FULLNAME>
<RETWEET_COUNT>0</RETWEET_COUNT>
<URL> http://twitition.com/p6q53</URL>
<HASHTAGS> #Twitition </HASHTAGS>
<MENTIONS> TwitterUserNameToReply </MENTIONS>
</DOC>
```

Microblog track addresses a search task comprising a set of 50 topics where the user information need is represented by the topic at a specific time. An example of topic is:

```
<top>
<num> Number: MB001 </num>
<title> BBC World Service staff cuts </title>
<querytime> Tue Feb 08 12:30:27 +0000 2011 </querytime>
<querytweettime> 34952194402811904 </querytweettime>
</top>
```

where:

- the `num` tag contains the topic number.
- The `title` tag contains the user'query representation.
- The `querytime` contains the timestamp of the query in a human and machine readable ISO standard form.
- The `querytweettime` tag contains the timestamp of the query in terms of the chronologically nearest tweet id within the corpus.

We did not use any future evidence or external resource, that is all runs were built using Tweets2011 corpus only, and removing from the corpus information generated after the query timestamp. We thus created 50 Terrier [9] indexes, one for each topic (each index contains just tweets with a `<DATE>` value less than the `<querytime>` value associated to the related topic). The Terrier indexes have been created using the out-of-the-box indexing settings and skipping the following fields:

```
TrecDocTags.skip=DATE,SCREENNAME,FULLNAME,RETWEET_COUNT,URL,MENTIONS,A,TO,LA
```

3 Methodology

We have used a methodology for both first pass and query expansion which is free from parameters.

3.1 Ad-hoc component

We have defined an information retrieval model which is based on the inner product of two information measures.

For each posting (term-message pair) there are three random variables: the size of the message l, the relative frequency of the term in the message $\hat{p} = \frac{tf}{l}$, the frequency for an additional unit of information $\hat{p}^+ = \frac{tf+1}{l+1}$, where tf is the frequency of the term in the message, the prior probability p of the term, i.e. the relative frequency $\frac{\sum_t \sum_d tf}{\sum_d l}$ of the term in the collection. The term-message weight is

$$l \cdot \hat{p} \cdot \log_2 \frac{\hat{p}}{p} \cdot \log_2 \frac{\hat{p}^+}{\hat{p}}$$

In information theory $\sum_t \hat{p} \cdot \log_2 \frac{\hat{p}}{p} = D(\hat{p}||p)$ is the Kullback-Leibler divergence. The quantity $\log_2 \frac{\hat{p}}{p}$ is the additional coding cost of *one* token term in bits, when observing the true probability $\hat{p}$ instead the term prior p. Similarly, the quantity $\log_2 \frac{\hat{p}^+}{\hat{p}}$ is the additional coding cost of a single token with respect to the optimal encoding of the message when this token is added to the message. Therefore the quantity $l \cdot \hat{p} \cdot \log_2 \frac{\hat{p}}{p}$ is the additional coding cost of the term in bits, when observing the true probability $\hat{p}$ instead the term prior p.

The query-message function is the inner product of these two information measures and is called KLIM (for Kullback-Leibler based product of Information Measures):

$$l \cdot \sum_{t \in q} \hat{p} \cdot \log_2 \frac{\hat{p}}{p} \cdot \log_2 \frac{\hat{p}^+}{\hat{p}}$$

It can be shown that this model is not statistically significant different from all main information retrieval models also on large TREC collections, like GOV2.

3.2 Query Expansion Component

We have used the parameter-free model Bo1 of query expansion [1] (QE). We have considered the first 30 messages in the ranking to expand the original query up to 10 new weighted query terms. The parameter free QE methodology is explained in [1] and is the out-of-the-box QE component of Terrier [9].

3.3 Dynamic cut Component

In order to predict the optimal cut for the retrieval messages, we analyzed the distribution of the most recent documents in the retrieved set. Our assumption is that the optimal cut value should keep the most relevant (with highest scores) messages in the retrieval set as much as possible, together with the most recent ones. Due to the evaluation of the retrieval set with P@30, as official evaluation measure, our retrieval set was made up of the first 30 documents in the relevance ranking.If a very recent message is too far from the early positions in the relevance ranking, then it is risky to include it in the final ranking. As a consequence, we introduced a filtering approach based on a comparison of score and timestamps of the retrieved messages. The main idea is to choose a particular score value as filtering threshold. We observed that the most recent retrieved message is always before the 200th position in the ranking. Then the threshold is defined as the mean value of the scores of the 200th and the 30th messages.

3.4 Time Re-Ranking Component

As anticipated, we have assumed that time and relevance are two dependent variables. However, time and relevance yield two independent rankings, that need to be merged: the relevance score ranking, and the recency ranking, i.e the ranking obtained by reordering the retrieved messages by creation time. Each retrieved message has three values: the relevance score (score_r), the relevance rank (rank_r) and the recency rank (rank_t).

The approach described below was also successfully used in a different retrieval context, that is opinion retrieval [2].

Boosting relevance by time. Zipf's law easily connects ranks and probabilities, and a straight application of Zipf's law provides a simple but effective methodology to combine recency and relevance scores. The hypothesis is that relevance depends on time and is obtained as posterior probability by Bayes' Theorem:

$$p(q|time, m) = \frac{p(time|q, m) \cdot p(q|m)}{p(time|m)}$$

where q is the query and m is the observed message. Zipf's law is then applied to the ranking by time on the retrieved set. The rank-frequency relationship (Estoup-Mandelbrot-Zipf's law) on the retrieved set is:

$$p(time|q, m) \sim \frac{\alpha_t \cdot B_t^{\alpha_t}}{(\text{rank}_t + B_t)^{(\alpha_t+1)}} \text{ with } \alpha_t > 0$$

where rank_t is the position of the message in the retrieved set re-ordered by timestamp, α_t is in general a very small constant, B_t is a parameter that needs to be learned. The factor α_t does not affect the ranking and can be deleted, moreover $\alpha_t \sim 0$:

$$p(time|q, m) \propto \frac{1}{\text{rank}_\tau + B_t au}$$

Assuming $p(q|m) \propto \text{score}_r$ and since $p(time|m)$ does not depend on the query, and is thus a constant, then we have the new ranking formula

$$p(q|time, m) \propto p(time|q, m) \cdot p(q|m) = \frac{p(q|m)}{\text{rank}_t + B_t} \propto \frac{\text{score}_r}{\text{rank}_t + B_t}$$

Thanks to the application of Zipf's law that establishes the relationship between scores and ranks, the combination of relevance score and recency rank is[1]:

$$\text{score}_{t,r} = \frac{\text{score}_r}{B_t + \text{rank}_t}$$

The final score is a normalization of the initial relevance score score_r by recency rank, rank_t. Time is used to obtain a moderate shifting of documents around their original positions, without thus performing a drastic permutation of the relevance score ranking. The *de facto* assumption is that the relevance score must be preferred to time for the final ranking.

At the time of run submissions, we did not have any training data to learn the parameters α_t and B_t of our approach so that we had to apply a heuristics to guess an acceptable value. As we said we made the hypothesis that the distribution was strictly Zipfian with $\alpha = 0$ (and indeed the value 0 for α_t was fully confirmed by experiments after the relevance data were available). In order to set B_t, note first that if $B_t \to \infty$ then the final ranking would become the relevance one. Therefore, the larger B_t is the less the changes to the relevance ranking. Instead of using an arbitrarily large value we used a rank-based constant that becomes large as long as the position grows in the ranking, i.e. $B_t = k \cdot \text{rank}_r$ and $k = 2$. In this way we knew in advance that possible shiftings of documents would have affected the upper part (and thus the early precision) of the ranking and not the tail of the ranking. Indeed, as shown on Tables 3 and 2 we can observe an improvement of both $MAP@_iX$ and $P@_iX$ (standard precision at X included) with any i and for the first positions $X \leq 8$. This setting describes the run DFReeKLIMRA.

In Section 5 we conducted an empirical study to assess the optimal value on the released training data, that is $B_t \sim 5000$. This value for B_t largely improves official results, in particular for $P@30$, but also for all $X \leq 1000$.

[1] We suggest to use $\text{score}^*_{t,r} = \frac{B_t \cdot \text{score}_r}{B_t + \text{rank}_t}$ for numerical reasons being $\text{score}^*_{t,r} = \text{score}_r$ for very large B_t.

4 Task objective and evaluation of the official runs

The main official evaluation measure that should be used to evaluate the runs is the precision at 30 ($P@30$). However, we here use a variant of the mean average precision measure as actual indicator of the quality of the ranking. The reasons why we have slightly modified both mean average precision at depth K, MAP@X, and precision at depth X, P@X, are for some consistency properties that their relationship should satisfy.

We recall that the average precision AP for a total ordering of the document collections is defined as

$$AP = \frac{1}{R} \sum_{r=1}^{R} \frac{r}{l(r)}$$

where $l(r)$ is the position of the r-th relevant document in the ordering and R is the number of relevant documents. The ranking is optimal when $r = l(r)$ for every $1 \leq r \leq R$, and in such a case AP is equal to 1. We may adapt AP truncating the ordering either at a fixed position, say X, or to a query-dependent position, say R, that is the total number of relevant documents for a query q. In order to generalize consistently the AP measure truncated at a certain position X, we need only to pay attention whether AP truncated at X may (or not) achieve 1 with the optimal ranking. AP truncated at X reaches 1 with optimal ranking with the following definition:

$$AP@_1X = \frac{1}{M} \sum_{r=1}^{r(X)} \frac{r}{l(r)}$$

where M is the minimum between X and R and $r(X)$ is the number of relevant documents that were retrieved in the first X positions (i.e. $l(r) \leq X$). If we instead use

$$AP@_2X = \frac{1}{R} \sum_{r=1}^{r(X)} \frac{r}{l(r)}$$

as definition of AP truncated at the X-th position, then we also consider the query difficulty factor. For difficult queries (containing in general many relevant documents) AP could be very low and in general for $R > X$ it cannot ever achieve the value 1.

A similar consideration holds for the other precision measures, such as the R-Precision, $P@R$, and the Precision at the Xth retrieved document, $P@X$. The standard R-precision can be regarded the positionless analogue [2] of the $AP@_1R$ (with $X = R$):

$$P@_1R = \frac{r(R)}{R}$$

where $r(R)$ is the number of relevant document retrieved in the first R positions. Note that $P@_1R$ is an upper bound for $AP@_1R$, that is *any permutation of documents applied only to the first R positions* gives $P@_1R \geq AP@_1R$, and $P@_1R = AP@_1R$ if and only if the $r(X)$ relevant and retrieved documents occupy the first R positions.

Analogously to AP we can generalize the notion of R-precision truncated at X retrieved documents. Note first, that the standard definition of precision at depth X, $P@X$, is defined as

$$P@X = \frac{r(X)}{X}$$

[2] As if the optimal ranking quality condition $r = l(r)$ were always satisfied for the relevant documents in the first R positions.

Table 1. A comparison of the official evaluation measure, $P@30$, to non-official ones, $P@_1 30$ and $MAP@_1 30$, on ranking by score (Relevance) and by time (Time). Note that the relevance scores of DFReeKLIMRA and DFReeKLIMZipf already include a time re-ranking component.

	DFReeKLIM		DFReeKLIM30		DFReeKLIMDC		DFReeKLIMRA		DFReeKLIMZipf
	Relevance	Time	Relevance	Time	Relevance	Time	Relevance	(Time)	Relevance
$P@30$	0.4395	0.1170	0.4401	0.4401	0.4395	0.3939	0.4476	(0.3918)	0.4537
$P@_1 30$	0.5619	0.1932	0.5626	0.5626	0.5619	0.4958	0.5677	(0.4951)	0.5761
$MAP@_1 30$	0.4136	0.1269	0.4138	0.4094	0.4136	0.3356	0.4192	(0.3193)	0.4311
$P@_1 30 - MAP@_1 30$	0.1483	0.0663	0.1488	0.1532	0.1483	0.1602	0.1485	(0.1758)	0.1450

However, we may alternatively define $P@X$ as a generalization of the R-precision with the truncation of the ranking at position X, that is:

$$P@_1 X = \frac{r(X)}{M} \text{ or } P@_2 X = \frac{r(X)}{R}$$

$P@_1 X$ can achieve 1 under optimal retrieval results, whilst both $P_2@X$ and standard $P@X$ may not achieve 1 even with no errors in the ranking[3].

It is easy to prove that $AP@_i X \leq P@_i X$, being $\frac{r}{l(r)} \leq 1$, but $AP@_i X \leq P@X$ for $i = 1, 2$ does not always hold. Therefore $P@_i X - AP@_i X$ for $i = 1, 2$ can be regarded as the error in precision of the ranking. In other words, for a given recall of the system the error of the precision in the ranking is $P@_i X - AP@_i X$, whilst $1 - P@_1 X$ is the margin for possible improvement for the retrieval set. However, the *official mean average precision of TREC evaluation tool is* $MAP@X = MAP@_2 X$ but $P@X$ is different from $P@_2 X$.

In the following, we use both $MAP@_1 X$ (the mean $AP@_1 X$ over all queries) and the difference $P@_1 X - MAP@_1 X$ as error rate of the system to assess the quality of the document ranking with truncation for our official runs.

4.1 Discussion of the official results

From Tables 1, 3 and 2 we derive the following conclusions:

a) The best cut value K to filter messages for any of the evaluation measures $MAP@_1 X$, $P@_1 X$ or $P@X$, is X itself. This result holds independently from whether none or some time re-ranking strategy is applied, as shown in the case $X = 30$ by the run DFReeKLIM30. More importantly, a pure truncation of the original ranking at depth $X = 30$, that is the run DFReeKLIM30, shows that $MAP@_1 30$ does not deteriorate significantly after reordering by timestamp the first 30 messages only ($MAP@_1 30$ decreases from 0.4145 to 0.4100). Therefore small improvements are expected by re-ranking DFReeKLIM30 (that is the first 30 messages of the baseline DFReeKLIM) by both relevance and time. Unless we explicitly assume the dependence of relevance to relevance (see DFReeKLIMRA below).

b) The dynamic cut strategy of DFReeKLIMDC for the K value did not work, since it worked for 14 queries out of the 49 queries.

c) Although we have observed with the remark a) above, that only small improvements can be obtained by re-ranking the given retrieval set ($X = 30$) by both relevance and time, the dynamic cut and time re-ranking strategy of the run DFReeKLIMRA improved the baseline (it increases from the baseline 0.4145

[3] It is sufficient that $X < R$ or $R < X$ respectively.

Table 2. DFReeKLIMRA and DFReeKLIMZipf include a time re-ranking component. DFReeKLIMRA improves early precision of the baseline without time re-ordering. Recency is thus a latent relevance indicator.

	DFReeKLIM	DFReeKLIMRA	DFReeKLIMZipf		DFReeKLIM	DFReeKLIMRA	DFReeKLIMZipf
	Relevance	Relevance	Relevance		Relevance	Relevance	Relevance
$MAP@_1 1$	0.6327	0.7551	0.6735	$P@_1 1$	0.6327	0.7551	0.6735
$MAP@_1 2$	0.5918	0.6939	0.6429	$P@_1 2$	0.6327	0.7143	0.6735
$MAP@_1 3$	0.5669	0.6355	0.6122	$P@_1 3$	0.6190	0.6735	0.6531
$MAP@_1 4$	0.5553	0.6003	0.5816	$P@_1 4$	0.6122	0.6480	0.6276
$MAP@_1 5$	0.5402	0.5780	0.5749	$P@_1 5$	0.6051	0.6398	0.6337
$MAP@_1 6$	0.5351	0.5521	0.5616	$P@_1 6$	0.6037	0.6207	0.6276
$MAP@_1 7$	0.5199	0.5294	0.5368	$P@_1 7$	0.5974	0.5986	0.5974
$MAP@_1 8$	0.5114	0.5106	0.5179	$P@_1 8$	0.5952	0.5859	0.5901
$MAP@_1 9$	0.4962	0.4935	0.5080	$P@_1 9$	0.5822	0.5714	0.5896
$MAP@_1 10$	0.4886	0.4822	0.4966	$P@_1 10$	0.5759	0.5653	0.5823

to 0.4198). Because of the remark a), the improvement is due to shifting upwards relevant documents with time. Although this is a positive evidence that time is a relevance indicator, it is still insufficient to show that recency is a decisive relevance boosting factor. With additional experiments and learning the unique parameter of the time component of the retrieval model, we can claim that time is a relevance boosting factor (see Section 5).

d) The difference $P@_1 30 - MAP@_1 30$ is constant in all the relevance rankings. Consequently, and as observed before, the improvement obtained by DFReeKLIMRA is not due to a better quality of the re-ordering by time, but to an effective boosting of relevance, that is able to shift homogeneously relevant documents upwards in the ranking.

e) Finally query expansion worked very well, and this shows that the task is very ad hoc.

4.2 How time affects relevance: an early precision analysis

Since there is not much difference between $P@_1 i$ and $MAP@_1 i$, with $i = 1, \ldots, 10$, for the baseline DFReeKLIM, we can deduce that the ranking by time of a very large retrieval set is almost optimal, in the sense that when we select the most recent messages from the retrieval set, the relevant messages are fewer than those in the relevance score ranking but always occupy the highest positions in time ranking. Therefore, the filtering strategy needs to take into account also not only recency but relevance. DFReeKLIMRA introduces a preliminary methodology based on the application of the Zipf law on the set of retrieved messages. Table 3) shows that DFReeKLIMRA improves the early precision of DFReeKLIM30 which is instead based on a pure reordering by time of the first 30 retrieved messages.

5 Evaluation of the non-official run

We have just one more non-official run to introduce, that is DFReeKLIMZipf, in order to complete the evaluation of our study. The time re-ranking component has the Zipfian parameter B_t that can be now learned trough the TREC evaluation data. For the new run DFReeKLIMZipf we have not here used cross-validation and results on Tables 1, 3 and 2 refers to the best matching value for B_t that is also for all defferent evaluation

Table 3. Recency affects relevance. DFReeKLIMRA improves the early precision of DFReeKLIM30 boosting relevance by recency.

	DFReeKLIM	DFReeKLIM30	DFReeKLIMRA	DFReeKLIZipf
	Time	Time	Relevance	Relevance
$P@_1 1$	0.7143	0.7755	0.7551	0.6735
$MAP@_1 1$	0.7143	0.7755	0.7551	0.6735
$P@_1 2$	0.4592	0.6327	0.7143	0.6735
$MAP@_1 2$	0.4541	0.6276	0.6939	0.6429
$P@_1 3$	0.3605	0.6020	0.6735	0.6531
$MAP@_1 3$	0.3458	0.5714	0.6355	0.6122
$P@_1 4$	0.3112	0.5816	0.6480	0.6276
$MAP@_1 4$	0.2925	0.5344	0.6003	0.5816
$P@_1 5$	0.2745	0.5857	0.6398	0.6337
$MAP@_1 5$	0.2505	0.5170	0.5780	0.5749
$P@_1 6$	0.2670	0.5884	0.6207	0.6276
$MAP@_1 6$	0.2312	0.5010	0.5521	0.5616
$P@_1 7$	0.2549	0.5797	0.5986	0.5974
$MAP@_1 7$	0.2147	0.4850	0.5294	0.5368
$P@_1 8$	0.2364	0.5782	0.5859	0.5901
$MAP@_1 8$	0.1987	0.4795	0.5106	0.5179
$P@_1 9$	0.2285	0.5680	0.5714	0.5896
$MAP@_1 9$	0.1892	0.4683	0.4935	0.508
$P@_1 10$	0.2201	0.5660	0.5653	0.5823
$MAP@_1 10$	0.1796	0.4623	0.4822	0.4966

measures. Table 1 shows that reordering DFReeKLIM by time DFReeKLIMZipf, and truncating it at depth 30, we would have a $P@30 = 0.4537$, greater than our best run DFReeKLIMRA[4]. However, a stronger result hold: DFReeKLIMZipf improves relevance-based measures of DFReeKLIM and DFReeKLIMRA at any depth $X \leq 1000$.

The boosting of effectiveness by time of the run DFReeKLIMZipf shows a high plausibility of the conjecture that the distribution of the relevant documents re-ordered only by time is Zipfian.

6 Conclusions

We have indexed the html Tweets2011 corpus. We have used a new parameter free model of IR (KLIM) and the out-of-the-box parameter free query expansion methodology of Terrier to retrieve the first 1000 messages for each query (run DFReeKLIM). Then we have just cut the ranking of DFReeKLIM to at depth X with two approaches: $X = 30$ (DFReeKLIM30) and with a dynamical cut with a mean cut $X = 73$ (DFReeKLIMDC). Finally we have assumed that relevance is dependent on time, and that in the reordering of the DFReeKLIM by time, relevant documents fit a Zipfian distribution. Under this assumption we have indeed improved relevance-based measures but at early precision, i.e. P@X with $X \leq 8$ (DFReeKLIMRA).

With the acquisition of the official relevance data of TREC assessment, we tuned the unique parameter B_t of the Zipfian distribution. We obtained a new run (DFReeKLIMZipf) that improves DFReeKLIMRA

[4] DFReeKLIMRA is obtained with a dynamic truncation that did not work, and this is the reason why $P@30 = 0.3912$ and not 0.4476 that would be obtained if the cut were done at depth 30.

at any depth $X \leq 1000$. Therefore we have proved that time can provide a uniform boosting of relevance. This outcome confirms a similar result obtained with a different collection (the Blog06 collection), where we applied query expansion selecting the pseudo relevant set with time distribution over documents [4]. Also in this work, time was shown to improve the performance of the first pass retrieval.

References

1. Giambattista Amati. *Probability Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, University of Glasgow, June 2003.
2. Giambattista Amati, Edgardo Ambrosi, Marco Bianchi, Carlo Gaibisso, and Giorgio Gambosi. Automatic construction of an opinion-term vocabulary for ad hoc retrieval. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen W. White, editors, *ECIR*, volume 4956 of *Lecture Notes in Computer Science*, pages 89–100. Springer, 2008.
3. Giambattista Amati, Giuseppe Amodeo, Valerio Capozio, Carlo Gaibisso, and Giorgio Gambosi. On performance of topical opinion retrieval. In Fabio Crestani, Stéphane Marchand-Maillet, Hsin-Hsi Chen, Efthimis N. Efthimiadis, and Jacques Savoy, editors, *SIGIR*, pages 777–778. Acm, 2010.
4. G. Amodeo, G. Amati, and G. Gambosi. On relevance, time and query expansion. In Craig Macdonald, Iadh Ounis, and Ian Ruthven, editors, *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 1973–1976. ACM, 2011.
5. Gordon V. Cormack, Ondrej Lhotak, and Christopher R. Palmer. Estimating precision by random sampling (poster abstract). In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 273–274, New York, NY, USA, 1999. ACM.
6. Craig Macdonald, Iadh Ounis, and Ian Soboroff. Overview of the TREC 2007 blog track. In Ellen M. Voorhees and Lori P. Buckland, editors, *TREC*, volume Special Publication 500-274. National Institute of Standards and Technology (NIST), 2007.
7. R. Manmatha, T. Rath, and F. Feng. Modeling score distributions for combining the outputs of search engines. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 267–275, New York, NY, USA, 2001. Acm.
8. Douglas W. Oard, Björn Hedin, Stephen Tomlinson, and Jason R. Baron. Overview of the TREC 2008 legal track. In Ellen M. Voorhees and Lori P. Buckland, editors, *TREC*, volume Special Publication 500-277. National Institute of Standards and Technology (NIST), 2008.
9. I. Ounis, G. Amati, Plachouras V., B. He, C. Macdonald, and D. Johnson. Terrier Information Retrieval Platform. In *Proceedings of the 27th European Conference on IR Research (ECIR 2005)*, volume 3408 of *Lecture Notes in Computer Science*, pages 517 – 519. Springer, 2005.
10. C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948.
11. Wolfgang G. Stock. On relevance distributions: Brief communication. *J. Am. Soc. Inf. Sci. Technol.*, 57:1126–1129, June 2006.