On the Impact of Entity Linking in Microblog Real-Time Filtering

Giacomo Berardi, Diego Ceccarelli, Andrea Esuli and Diego Marcheggiani
Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
via Giuseppe Moruzzi, 1, 56124 Pisa, Italy
firstname.lastname@isti.cnr.it

ABSTRACT

Microblogging is a model of content sharing in which the temporal locality of posts with respect to important events, either of foreseeable or unforeseeable nature, makes applications of real-time filtering of great practical interest.

We propose the use of Entity Linking (EL) in order to improve the retrieval effectiveness, by enriching the representation of microblog posts and filtering queries. EL is the process of recognizing in an unstructured text the mention of relevant entities described in a knowledge base. EL of short pieces of text is a difficult task, but it is also a scenario in which the information EL adds to the text can have a substantial impact on the retrieval process.

We implement a start-of-the-art filtering method, based on the best systems from the TREC Microblog track real-time adhoc retrieval and filtering tasks , and extend it with a Wikipedia-based EL method. Results show that the use of EL significantly improves over non-EL based versions of the filtering methods.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithm, Experimentation

Keywords

Real-time filtering, Microblogging, Entity Linking

1. INTRODUCTION

Microblogging has gained great popularity in the last years, with Twitter leading the field with about 500M of tweets per day and 255M of monthly active users[1]. Novice users can be overwhelmed by such a sheer amount of information, and even technically skilled users can have a hard time when searching for some specific information, especially about recent or currently happening events. For these reasons the real-time filtering problem recently emerged as a relevant IR problem, as shown by the TREC Microblog Track [23].

The task of microblog real-time filtering starts with just a query and then a time-sorted stream of microposts (tweets) must be processed filtering out tweets that are non-relevant with respect to the query. The real-time aspect of the process characterizes the problem by (i) imposing limits on the computational cost of the proposed solutions, and (ii) generating a stream of relevance feedback on the tweets that are marked as relevant. The filtering component of the search system must be able to quickly take the filtering decision on each micropost, avoiding to be a bottleneck in the presentation of search results to the user. The filtering process starts with very little relevance information; the feedback that accumulates during the stream processing is a crucial information that the filtering method can exploit, if it can do it efficiently, to tune its parameters on the fly.

In this paper we propose and investigate the use of Entity Linking (EL) to enrich microposts and queries representations in order to improve the filtering process. The goal of EL is to recognize in an unstructured text the mention of relevant entities described in a knowledge base. Wikipedia[2] is usually adopted as the knowledge base, with each of its pages denoting an entity. For example, the mentions of "TREC" in this paper can be linked to the entity denoted by the "Text Retrieval Conference" page[3] in Wikipedia.

The purpose of using EL in microblog retrieval is mainly toward improving recall, as EL can link together the various ways an entity can be mentioned. For example, EL enables to connect the query "Michael Schumacher health conditions", to a relevant piece of text in which the entity is mentioned using a different expression "Yes! Schumi is out of the coma!". In short pieces of text such as microposts it is likely to expect a preference for shorter versions of names, just to fit within space limitations, or for names that are more pertinent with respect to the topic of the post, e.g., preferring "French President" or "François Hollande" when either referring to the public role or to the personal life events, but no strong assumptions can be made, specially in the case of Twitter in which almost every relevant tweet has a distinct author.

A secondary purpose of EL in microblog retrieval can be

[1] http://goo.gl/46oMx9 http://goo.gl/J05X3D

SAC 2015 April 13-17, 2015, Salamanca, Spain.

ACM 978-1-4503-3196-8/15/04 $15.00.
http://dx.doi.org/10.1145/2695664.2695761.

[2] http://wikipedia.org
[3] http://en.wikipedia.org/wiki/Text_Retrieval_Conference

toward improving precision, as EL can help to solve ambiguities in text. However, EL methods usually rely on the presence of mentions of many different entities in the piece of text being analyzed in order to resolve the ambiguities that some of the mentions may present, and short texts such as microposts may often contain an insufficient number of mentions to allow the method to solve the ambiguities. Moreover, in order to perform a complete disambiguation, some EL methods require a relatively costly graph-based computation for each processed text, which makes impossible to use them in a real-time setup.

We implement a state-of-the-art filtering method, based on the best systems from the TREC Microblog real-time filtering track, and extend it with a Wikipedia-based EL method. Results of comparative experiments show that EL-aided filtering methods obtain a significantly better performance over non-EL methods. Our solutions are effective but also efficient, e.g., they can run on a personal device that gets an input stream, via API, from a microblog infrastructure.

2. PROBLEM DESCRIPTION

The task of real-time filtering, as described in the TREC competition, consists in filtering a time-sorted stream of tweets $S = \{s_1, s_2, ..., s_n\}$, by classifying each one as either relevant or non-relevant with respect to a given query q. In the case a tweet $s_t \in S$ is classified as relevant for the query, it is possible to get a relevance feedback stating the correct relevance judgment of s_t. The relevance feedback can be used to update the filtering model for the classification of tweets that have been posted after the time t.

2.1 A Supervised Filtering Approach

Following the current state-of-the-art system for the task of real-time filtering [1], we adopt a supervised approach well-known for its effectiveness in adapting filtering tasks (e.g., [26]), namely Incremental Rocchio classifier [2]. Incremental Rocchio allows to create a user profile using both relevant and not relevant documents, tweets in our case. Each time a new tweet is proposed to the user, Incremental Rocchio checks whether the new tweet is relevant according to the user profile.

Formally, the user profile is calculated as a centroid $\vec{c}_t$ in the vector space:

$$\vec{c}_t = \frac{\alpha}{|R_t|} \cdot \sum_{s_i \in R_t} \vec{s}_i - \frac{\beta}{|N_t|} \cdot \sum_{s_i \in N_t} \vec{s}_i \qquad (1)$$

Where R_t is the set of relevant tweets at time t, N_t is the set of non-relevant tweets a time t, α and β are parameters that weight the "importance" of relevant and non-relevant tweets respectively. $\vec{s}_i$ represents the vectorial transformation of the tweet s_i. The index t indicates the time-step in the streaming S; given the possibility of a relevance feedback, the centroid $\vec{c}_t$ can change through time.

In order to check if a new tweet s_t is relevant for the query, the cosine distance between the centroid $\vec{c}_t$ and the vectorial representation of the tweet $\vec{s}_t$ is calculated. If the distance is lower than a certain threshold η (determined on a validation set, see Section 5) the tweet is classified as relevant, otherwise as non relevant. In the case in which the tweet is classified relevant a relevance feedback is obtained. If the tweet is actually relevant it will be added to the set R_{t+1} otherwise it will be added to the set N_{t+1}. The centroid $\vec{c}_{t+1}$ is then updated to take into account the new information.

Following the guidelines of the microblog track, at the beginning of the filtering process R_0 is composed by the query and the first relevant tweet.

2.1.1 Features

In our reference system, which does not use EL, we convert tweets to their vectorial representation by extracting a variety of features. We take as a feature each word (stopwords are removed), and also its stemmed version, obtained from the Lancaster stemmer [19], which resulted to be more effective than other stemmers. Stems are marked as distinct features from words that have the same spelling, e.g., "run" and "stem:run", in order to avoid unwanted alterations of word frequencies. We take word bigrams by using a sliding window on text; stopwords are not removed in this case. The title of Web pages pointed by the URLs appearing in tweets may contain information that is useful to determine their context and thus their relevance; we retrieve titles from linked Web pages and we extract the same features as above. Hashtags may be composed of words that are useful to determine relevance; we perform Viterbi algorithm-based hashtag segmentation [3], e.g., "#royalvisitusa" becomes "royal visit usa". Hashtags are added as features in both original and segmented forms. We use $tf \cdot idf$ weighting, which, for our filtering method, resulted the best performer in experiments on the validation set comparing various weighting models proposed for the track.

2.1.2 Filtering

Han et al. [9] observed that the presence of a URL, which may either point to a Web page or an image, in a tweet is a relevant hint for the relevance of a tweet. For example, 84% of the set of relevant tweets in the validation set contain at least a URL, while this value is only 23% for non-relevant ones. In [9] a large boost in precision, with a symmetric loss in recall, has been reported by considering as relevant only tweets that contain at least a URL (the "hitUWT" run). We tested two versions of the reference system, one that only uses the Incremental Rocchio classifier (IncRoc in Table 1), and one that marks as non-relevant any tweet that does not contain a URL (IncRocU in Table 1)

For the binary classification by the Incremental Rocchio classifier, the centroid is initially determined on the query vector $\vec{q}$ and the vector of the first relevant tweet of the stream, which is given for each query of the dataset. The *idf* weights are initially computed on the last 1000 tweets before the first relevant tweet. The costs of the centroid update and the cosine similarity is linear with the size of the centroid (i.e., the size of the set of features appearing in relevant tweets), when using efficient sparse data structures and keeping a dictionary of the *idf* weights of the features.

3. ENTITY LINKING

EL allows to connect small text fragments in a document with entities contained in a given knowledge base, e.g., Wikipedia. Given a plain document the linking is usually performed in 2 steps: the text fragments that could refer to an entity (called *spots*, *mentions*, or *surface forms*) are identified. Since a mention may represent several different *entities* (e.g., the mention "Italy" can refer to the country or the Italian football team), a *disambiguation* step is performed, where the correct entity is selected among the candidates.

Let us define $M = \{m_1, m_2, \ldots, m_{|M|}\}$ and $E = \{e_1, e_2, \ldots, e_{|E|}\}$ as the sets containing respectively all the mentions and the entities (articles) of Wikipedia. Each mention links to one or more entities. In this work we rely on two important measures used for EL: the *link probability* and the *commonness*. Given a mention m, $lp(m)$, the link probability represents the probability of the text m to be a link to an entity in Wikipedia. This property allows us to discriminate mentions that link, with a high probability, to some entity from those referring to an entity only occasionally. For example, the mention "the" occurs a huge number of times in Wikipedia, but only in a few cases links to the Wikipedia page about the English articles. Given an entity e and a mention m, $cm(m, e)$ represents the commonness, i.e., the probability $p(m|e)$ that a mention m links to e. For each mention, the sum of the commonness of all its related entities is 1. The commonness determines the strength of the relation mention-entity. These values are computed on the Wikipedia collection used by the entity linker.

4. ENTITY LINKING-AIDED REAL-TIME FILTERING

As already discussed, the main purpose of using EL in filtering is to increase the possibility of linking semantically related information expressed in different forms.

The text of queries, tweets, and the title of Web pages linked by tweets, are analyzed by an entity linker that produces a list of found mentions, each one paired, with a link probability value, with a *candidate entity*. We also define as the *surface forms set* (SF) of an entity e the set of mentions $SF(e) = \{m \mid cm(e, m) > 0\}$, i.e., all the mentions that link to the entity e at least once. For example, the surface form set for the entity *Diego Armando Maradona* (DAM in the following), is constituted by the mentions "argentine legend", "diego armando maradona", "el diego", etc.

It is worth noting that in our efficiency- and recall-oriented approach, we do not perform disambiguation based on the relations among the candidate entities in the knowledge base graph. Our approach mainly relies on commonness, and is inspired by the simple method evaluated by Meij et al. [14]. In that paper, Meij et al. prove that linking the entity with highest commonness with respect to the mention has a good performance on tweets. In particular they show that the commonness method applied on tweets outperforms the traditional state-of-the-art methods [8, 15, 16]. The authors of [14] also propose a machine learning approach to improve the performance of EL on tweets, but we decided to rely on *commonness* for its simplicity and also because it represents a good trade-off between quality and efficiency.

We explore three ways of expanding the features space in which tweets are represented by means of EL. Two of the methods leverage on the bipartite graph of mentions and entities. Navigating that graph is a critical process, since it is necessary to find a sweet point in the exploration range, in order to be able to find new relevant information without adding undesired noise.

4.1 Features expansion by highlighting mentions

This method (dubbed Exp1 hereafter) adds a new feature for each mention found in text. For example, for "el diego", it adds the feature "ment:el_diego". The rationale behind this method is to promote, by replicating it, the weight of mention-related components of text in the determination of similarities between vectors. A threshold for the minimum link probability for which a piece of text can be considered a relevant mention must be determined experimentally (see Section 5.3). Note that different surface forms for the same entity are not linked by this simple method, an issue addressed by the other two methods.

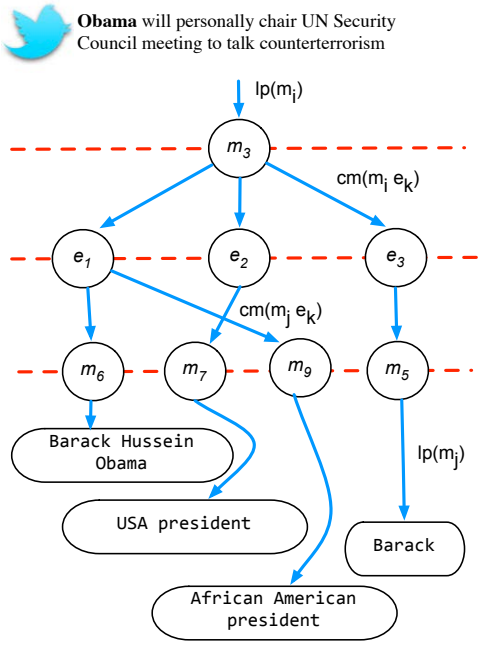

Figure 1: The directed acyclic graph used when computing feature expansion. In this example, "Obama" matches a mention m_3 that refers to several entities, each one linking to a number of other mentions. For each of these mentions the relative surface form is shown. The original mention and the surface forms found navigating the graph are connected through paths, in which edges are weighted by the lp and cm functions, as indicated.

4.2 Features expansion by exploring entities

In Figure 1 we represent an example of a graph generated with the information obtained by the entity linker. The nodes of the graph are mentions and entities, the edges are weighted by the probabilities given by lp and cm.

The second expansion method (dubbed Exp2) selects the most relevant entities returned by the entity linker for all the mentions in a tweet, and creates new features from them. We define $p_{en}(m_i, e_k)$, the probability of linking the mention m_i to the entity e_k, as:

$$p_{en}(m_i, e_k) = lp(m_i) \cdot cm(m_i, e_k) \quad (2)$$

Candidates entities are ranked by p_{en} and those with a score higher than a parameter ρ, optimized on a validation set, are selected. With Exp2 the mentions "el diego" and "Armando Maradona" are expanded into the same feature "ent:eDAM", i.e., the unique identifier of the entity, thus effectively enabling the possibility of linking two pieces of text in which

the same entities are mentioned using different mentions.

4.3 Features expansion by exploring surface forms

The third expansion method (dubbed Exp3) expands a tweet with all the alternative surface forms of the entities in the tweets. We define $p_{sf}(m_i, e_k, m_j)$ as the probability of associating the surface form m_j to the mention m_i through the entity e_k.

$$p_{sf}(m_i, e_k, m_j) = lp(m_i) \cdot cm(m_i, e_k) \cdot lp(m_j) \cdot cm(m_j, e_k) \quad (3)$$

where e_k is an entity which connects the mentions m_i and m_j. Similarly to Exp2, a parameter ρ sets the threshold for the selection of which candidates have to be expanded into new features.

In the case of Exp3 the example mention "el diego" is thus expanded into a set of features of all the surface forms m_j for the entity **DAM** for which $p_{sf}(\text{"el diego"}, DAM, m_j) > \rho$, e.g., "sf:armando_maradona", "sf:argentine_legend", *etc.*

With Exp3, the entities with more surface forms are likely to have a higher weight in the determination of similarity between vectors, due to the larger number of features generated. Depending on the $lp(m_i)$ and $cm(m_i, e_k)$ values for the original mention m_i from which p_{sf} is computed, the surface forms for an entity e_k gets different weights with respect to other initial mentions m_x. This results in different sets of features being added to vectors depending on the initial mention, thus adding a grading in the similarity between vectors referring to the same entity using different forms.

Moverover, ambiguous surface forms, such as "Formula One driver" may generate common features for pieces of text in which the original mentions were linked to different, but related, entities, e.g., creating a common feature from the entity "Nico Rosberg" and the entity "Lewis Hamilton"; this can however result in adding features that link unrelated pieces of text. The behaviour is somewhat expected from the Exp3, since it is the most aggressive method of the three; experiments will determine if the benefits from the additional linking possibilities are stronger or weaker than the errors introduced by ambiguous surface forms.

5. EXPERIMENTS

5.1 Dataset

We compared the methods described in Sections 2 and 4 on the dataset originally made available by TREC for the microblog ad-hoc retrieval task [18] and used also for the real-time filtering task [23].

Originally consisting of 16M tweets, due to Twitter policies it must be independently downloaded by each research group, resulting in difference as tweets and accounts are deleted. Our download resulted in 14M tweets retrieved. Similarly to [9], Table 1 reports the recall for the trivial acceptor filtering, AllRel, indicating the coverage of our copy of the dataset with respect to the original one.

The microblog track organizers also provided the relevance judgments for 49 queries. Ten queries form a validation set that can be used for parameter optimization. The other 39 queries form the test set on which we evaluate the methods.

5.2 Evaluation Measures

As evaluation measures we adopt the same four evaluation measures that have been chosen for the real-time filtering task of the microblog track: *precision*, *recall*, $F_{0.5}$ and $T11SU$. The $F_{0.5}$ function is the F_β function with $\beta = 0.5$, i.e., preferring *precision* over *recall*. The $T11SU$ is the *scaled linear utility*, a measure for the adaptive filtering task [21]. Results are averaged across the 39 test queries.

5.3 Experimental Setting

All the code of the real-time filtering system we implemented, together with the instructions for reproducing the experiments, is released with an open-source license and is available at `https://github.com/giacbrd/CipCipPy`. For the EL functionalities we used Dexter[4] [4] a versatile open-source framework for EL.

We have tuned the parameters of the system on the validation set, evaluating them with respect to the $F_{0.5}$ measure. We have optimized the parameters distinctly for each of the tested setup reported in Table 1. We have run grid search experiments to optimize the α and β parameters of Equation (1), along with the η threshold for the classifier. Results indicate that using non-relevant tweets has a negative impact and it is better to consider only relevant tweets (i.e., $\beta = 0, \alpha = 1$, confirming the similar findings of [1]). The actual centroid formula we used is thus:

$$\vec{c}_t = \frac{1}{|R_t|} \cdot \sum_{s_i \in R_t} \vec{s}_i \quad (4)$$

Another parameter to tune is ρ, which is used to exclude candidate features (either entity ids, or surface forms) with low probability; we have found that its optimal value is the same for all the entity expansion approaches, i.e., 0.1. We also determined that the value of minimum link probability that a mention should have, in order to be linked, is 0.2. This value is however not crucial for the efficacy of the expansion, which is instead strictly dependent on ρ.

5.4 Results

Table 1 reports the results of our methods and a number of methods we compare to. Medians2012 are the median values for all the participants to the 2012 TREC Microblog real-time filtering track; Best2012 is the best performer of that track. The current best system, with respect to $F_{0.5}$, in real-time filtering is CurrentBest, namely [1].

AllRel is a trivial acceptor system which marks all tweets as relevants; this allows to assess the coverage of relevant tweets of the our copy of the corpus, which is a good 95.4%[5]

IncRoc and IncRocU are our implementation of supervised method we describe in Section 2.1, inspired to the state-of-the-art systems. IncRocU differs from IncRoc as it follows the findings of [9] and it marks any tweet which does not contain a URL as non-relevant. Both system obtains very good scores, competitive with the state of the art. IncRocU obtains a higher recall and $F_{0.5}$ with respect to the state-of-the-art systems, and it is also the system which better balances precision and recall. Since IncRocU obtains the highest $F_{0.5}$ and $T11SU$ scores, it will be our strong reference system on which we test our expansion methods.

[4] `http://dexter.isti.cnr.it/`

[5] For comparison, [9] reports 91.5%.

Method	Precision	Recall	$F_{0.5}$	T11SU
Medians2012	.177	.334	.149	.208
Best2012	**.622**	.174	.334	**.412**
CurrentBest [1]	.421	.337	.344	.361
AllRel	.000	.954	.000	.000
IncRoc	.329	.423	.323	.284
IncRocU	.408	.382	.372	.357
Exp1	.417	.383	.383	.369
Exp2	.420	**.390**	**.389**	.372
Exp3	.371	.341	.342	.327
Exp2-1Ent	.409	.379	.377	.367

Table 1: Evaluations of the runs. In bold the best result for a specific evaluation measure.

Experiments that use the proposed EL-based expansion methods are listed as Exp1, Exp2 and Exp3. The Exp1 method (Section 4.1) obtains a relative +2.2% improvement in precision, with almost no variation in recall ($F_{0.5}$ and $T11SU$ respectively improve by +3.0% and +3.4%). This indicates that just giving more importance to mentions over the rest of the text allows to better differentiate relevant and non-relevant tweets, following the intuition that mentions of entities strongly characterize the content of tweets. The Exp2 method (Section 4.2) is the top performer, and improves on all the measures (Precision +2.9%, Recall +2.1%, $F_{0.5}$ +4.6%, $T11SU$ +4.2%). The significant increase in recall shows the impact of entity-based features in strengthening the similarities among documents which talk about the same concepts, possibly using different expressions. Precision increases, though marginally, with respect to Exp1, meaning that the potential noise or ambiguity generated by entity-based expansion is null or irrelevant. The Exp3 method (Section 4.3) instead worsens all the results with respect to the base system, indicating that a more aggressive expansion adds detrimental ambiguity and noise into the text.

We also tested a variation of the Exp2 expansion method, Exp2-1Ent, which adds for each mention identified by the linker only the candidate entity with the higher commonness. This variation follows the work of Meij et al. [14] (see Section 4), and can be considered as a light disambiguation step. Exp2-1Ent slightly improves over the base system (not on recall), but it is worse than Exp2, indicating that being more inclusive in adding entities is a better strategy.

To sum up, we started from a strong baseline as IncRocU, and by simply using information in the knowledge base to promote mentions to first class features (Exp1) we improved precision. When using a proper EL-based method to perform expansion by entities (Exp2), thus enabling linking similar concepts expressed in different forms, also recall, and all the other measures, improved.

6. RELATED WORK

IR in microblogging has been investigated in the works of Efron [6], Nagmoti et al. [17] and Weng et al. [24]. Efron overviews various IR tasks in the domain of microblogging. Weng et al. propose an extension of PageRank algorithm to measure the influence of users in microblog platform such as Twitter. Nagmoti et al. propose several ranking strategies for ad-hoc retrieval in the microblog domain.

The first large scale initiative that has focused on microblogging is the 2011 TREC Microblog Track [18], with the task of ad-hoc real-time search. The second edition of TREC Microblog Track [23] proposed also the real-time filtering task. Several research groups have participated to the real-time filtering task. Many of them have faced the filtering task adopting supervised learning methods, and binary classification is performed on tweets by employing several techniques. Online learning methods, such as the Rocchio algorithm, have been widely used [12, 13], but also batch learning techniques have been adopted [11, 12]. Among the top performing methods there also methods that are mainly based on the retrieval scores produced by IR systems designed for the ad-hoc search task. In particular, the system that ranked first [9] adopted a method designed for the ad-hoc retrieval task (based on learning to rank), performing a reference search on the documents preceding the query and judging any new tweet from the stream as relevant whether its retrieval score is not smaller than the scores of the m most relevant tweets from the reference search. In the same way, the system that ranked second [25] adopted an ad-hoc retrieval system based on learning to rank and, in order to face the filtering task, it implemented an adaptive threshold mechanism. A notable work that followed the 2012 TREC Microblog Track real-time filtering task is from Albakour et al. [1], which proposes a Rocchio-based system extended with a query expansion technique for dealing with sparsity.

EL is widely used in text mining, especially for short texts [8, 10]. Some recent works in EL [5, 20, 22] adopt modern approaches for expanding textual representations for queries and documents. They use the graph structures returned by the EL frameworks to obtain new features that are semantically related to the pieces of text being analyzed.

Use of EL in microblog ad-hoc retrieval has been proposed by Feltoni and Gasparetti [7]. Given a query, the top 15 results from a standard IR system are given in input to a wikification service; the annotated entities are then compared to the terms in the original query and those with the highest semantic relatedness, as determined by another service that leverages on the Wikipedia link graph, are then used for query expansion on the IR system. In addition to the inherent differences between the ad-hoc retrieval and real-time filtering tasks, our work differs from [7] as we do not use an IR system, we do not use EL output from tweets to expand the query, and our method automatically adapts its parameters based on relevance feedback. Our method is also computationally lighter, as it does not perform full wikification of text nor it uses any complex graph-based processing.

7. CONCLUSIONS

Real-time filtering of microblogs is a non-trivial task. It poses a combination of challenges which differentiate it from related tasks such as ad-hoc retrieval on microblogs or filtering from other sources of content: short texts, use of jargon/hashtags, real-time content evolution and feedback, compromises between responsiveness and efficacy.

We implemented a reference system that is competitive with (and for some metrics better than) the current state of the art. On top of that system we tested the impact of different ways of using EL to enrich the content of tweets. Two of the methods, Exp2 over all, considerably improved the non-EL results, showing how even a simple and efficient method for incorporating the information from an external

knowledge can have an impact on retrieval applications. We consider that our intuition of using EL to improve the filtering effectiveness has been confirmed by the results.

Even though the expansion methods were originally designed to improve recall, results have shown a sensible improvement in precision too, indicating that mentions and entities are a key element of content in order to determine its relevance with respect to a query.

Acknowledgements This work was partially supported the Regional (Tuscany) project SECURE! (POR CReO FESR 2007/2011)

8. REFERENCES

[1] M. Albakour, C. Macdonald, and I. Ounis. On sparsity and drift for effective real-time filtering in microblogs. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM'13)*, pages 419–428, San Francisco, US, 2013.

[2] J. Allan. Incremental relevance feedback for information filtering. In *Proceedings of the 19th annual international ACM SIGIR '96. Zurich, CH, 1996.*

[3] G. Berardi, A. Esuli, D. Marcheggiani, and F. Sebastiani. ISTI@ TREC Microblog track 2011: exploring the use of hashtag segmentation and text quality ranking. In *Proceedings of the 20th Text REtrieval Conference (TREC 2011)*, Gaithersburg, US, 2011.

[4] D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, and S. Trani. Dexter: an open source framework for entity linking. In *Proceedings of the 6h International Workshop on Exploiting Semantic Annotations in Information Retrieval, (ESAIR'13)*, pages 17–20, San Francisco, US, 2013.

[5] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'14)*, pages 365–374, Gold Coast, AU, 2014.

[6] M. Efron. Information search and retrieval in microblogs. *Journal of the American Society for Information Science and Technology*, 62(6):996–1008, 2011.

[7] D. Feltoni Gurini and F. Gasparetti. Trec microblog 2012 track: Real-time algorithm for microblog ranking systems. Technical report, 2012.

[8] P. Ferragina and U. Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10)*, pages 1625–1628, Toronto, CA, 2010.

[9] Z. Han, X. Li, M. Yang, H. Qi, S. Li, and T. Zhao. Hit at trec 2012 microblog track. In *Proceedings of Text REtrieval Conference*, 2012.

[10] X. Hu, N. Sun, C. Zhang, and T.-S. Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proceedings of CIKM 2009*, Hong Kong, CN, 2009.

[11] S. Karimi, J. Yin, and P. Thomas. Searching and filtering tweets: Csiro at the trec 2012 microblog track. Technical report, 2012.

[12] F. Liang, R. Qiang, Y. Hong, Y. Fei, and J. Yang. Pkuicst at trec 2012 microblog track. Technical report, 2012.

[13] N. Limsopatham, R. McCreadie, M.-D. Albakour, C. Macdonald, R. L. T. Santos, and I. Ounis. University of glasgow at trec 2012: Experiments with terrier in medical records, microblog, and web tracks. Technical report, 2012.

[14] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *Proceedings of the 5th ACM international conference on Web search and data mining (WSDM'12)*, pages 563–572, Seattle, US, 2012.

[15] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8, Graz, AU, 2011.

[16] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management (CIKM'08)*, pages 509–518, Napa Valley, US, 2008.

[17] R. Nagmoti, A. Teredesai, and M. De Cock. Ranking approaches for microblog search. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '10, pages 153–157, Washington, DC, USA, 2010. IEEE Computer Society.

[18] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff. Overview of the trec-2011 microblog track. In *Proceeddings of the 20th Text REtrieval Conference (TREC'11)*, 2011.

[19] C. D. Paice. Another stemmer. *SIGIR Forum*, 24(3):56–61, 1990.

[20] Z. Ren, M.-H. Peetz, S. Liang, W. van Dolen, and M. de Rijke. Hierarchical multi-label classification of social text streams. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'14)*, pages 213–222, Gold Coast, AU, 2014.

[21] S. Robertson and I. Soboroff. The trec 2002 filtering track report. In *Proceedings of the 11th Text REtrieval Conference (TREC'02)*, Gaithersburg, US, 2002.

[22] M. Schuhmacher and S. P. Ponzetto. Knowledge-based graph document modeling. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM'14)*, pages 543–552, New York, US, 2014.

[23] I. Soboroff, I. Ounis, C. Macdonald, and J. Lin. Overview of the trec-2012 microblog track. In *Proceedings of the 21st Text REtrieval Conference (TREC'12)*, 2012.

[24] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 261–270, New York, NY, USA, 2010. ACM.

[25] J. Zhang, S. Chen, Y. Liu, J. Yin, Q. Wang, W. Xu, and J. Guo. Pris at 2012 microblog track. Technical report, 2012.

[26] Y. Zhang and J. P. Callan. The bias problem and language models in adaptive filtering. In *TREC*, 2001.