WISTUD at TREC 2011 Microblog Track:

Exploiting Background Knowledge from DBpedia and News Articles for Search on Twitter

Ke Tao, Fabian Abel, Claudia Hauff

Web Information Systems, Delft University of Technology
{k.tao, f.abel, c.hauff}@tudelft.nl

Abstract. These working notes describe the system developed by the WISTUD team for the Microblog track. We evaluated the suitability of semantic technologies for the search task, in particular, query expansion with named entities that are deduced by means of a topic-based profiling process. The results indicate the feasibility of the approach: for half of the topics, at P@30, our top performing automatic method based on semantic profiling yields better results than the median over all submitted runs.

1 Introduction

In TREC 2011, we participated in the Microblog track, which was run this year for the first time. A corpus of sixteen million Twitter messages (so-called tweets) was released together with 50 search topics.

The search task is defined as retrieving the *interesting* and relevant tweets for a given topic and a given time frame. In this task, we consider the language gap between the topics and the Twitter messages as a core challenge. Since tweets are limited to a maximum of 140 characters, Twitter users often rely on abbreviations and memes to convey their message [1]. Given a query such as "Roger Federer Wimbledon 2009", we expect to find very few tweets that contain all query concepts. We found this indeed to be the case in this year's search topics: only 18 of the provided 50 search topics yield ten or more result tweets in the corpus when used as conjunctive queries. To overcome this problem, we implemented approaches that expand the original topics with keywords, that are more in line with the type of language people use on Twitter. Our solutions are inspired by Twitter-based user modeling methods [2] and aim to create a semantically rich profile for a topic that is then translated into a weighted keyword query. Therefore, each query is modelled as a list of weighted concepts. The concepts, e.g. *FIFA*, *Brazil*, *Barack Obama*, *Qatar*, *Sport*, and *Eurosport*, are automatically derived with a Named-Entity-Recognition service from (i) news articles by mainstream media outlets, and (ii) from a set of phrases commonly used in tweets, so-called memes. It should be stressed that although we used a news corpus for query expansion, due to time constraints, we did not consider any of the web pages linked to in Twitter messages as evidence. Apart from the automatic runs submitted based on these topic modelling strategies, we also

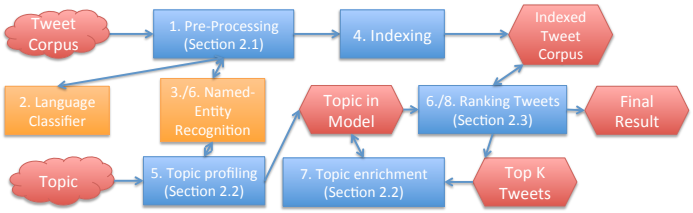

Fig. 1. Overview of the search framework.

created one manual run based on searches performed by a human annotator. While our automatic runs - as expected - performed less well than the manual run, we found our best automatic run to outperform the median result in 50% of the topics (P@30).

The remainder of the paper is organized as follows: the search framework is described in detail in Section 2. The experimental setup is described in Section 3, followed by the results (Section 4). Section 5 discusses the results and Section 6 concludes the paper.

2 Search Framework

We interpret the interestingness of a tweet as the informativeness and coverage from different informations sources, such as news media, experts, and politicians. For each of the search topics, we derive a topic profile, which contains the concepts important to the topic and their corresponding weights. The concepts are derived from the enrichment process. The final query (the topic profile concepts and their weights) is submitted to the underlying retrieval engine[1]. Returned is a ranked list of tweets, together with their corresponding probability scores of relevance. Since the result tweets - by definition of the task - are ranked according to their time stamp instead of their relevance scores (as is the case in standard ad-hoc retrieval tasks), we define a rank threshold K. Tweets with a relevance rank below K are discarded from the result list. Finally, the tweets remaining in the result list are ordered according to their time stamp from newest to oldest. Since up to 1000 tweets could be submitted for each topic, we appended the truncated result list with additional tweets which were older than the oldest tweet in the top K results.

An overview of our search framework is shown in Figure 1. It contains three types of components: main components (in blue), assisting components (yellow) and data (red). In the following sections, we describe the corpus preparation, the topic profiling and finally the search.

2.1 Corpus Pre-Processing

As the corpus is distributed as a list of Twitter message IDs, we first had to crawl[2] the tweets. Apart from the message text itself, we also gain some important meta-

[1] Indri, `http://http://www.lemurproject.org/`
[2] Twitter search API, `https://dev.twitter.com/docs/api/1/get/search`

data such as geographic locations, the number of times a tweet was re-tweeted, the users' self-introduction, etc.

In the pre-processing phase, we filter out all tweets of the corpus that are not identified as English since - by definition of the task - only English tweets will be judged for their relevance. This is a text categorization problem that can be solved with an N-gram based method [3]. The N-gram based method generates language profiles from a number of training documents of which the languages are already known. Each profile is a list of the most frequent N-grams in the documents. By measuring the similarity (or distance) between the language profiles and the N-grams of each tweet (URLs were removed from the tweets, letters occurring in sequence 4+ times were replaced with a single letter), the most likely language of each tweet can be found. For our experiments, we relied on an existing language detection library[3]. If a tweet was classified as being in English with a probability of at least 0.85, it was included in the reduced English-only corpus.

2.2 Topic Modeling

We model each topic as a *topic profile* which should provide an accurate and comprehensive summary of the topic. When users submit a query to search for information, they are often not able to include the specific information nugget they are looking for. For example, a user's intention behind submitting a query such as "2022 FIFA Soccer" may be to learn which country has won the bid for the 22^{nd} FIFA World Cup. If the user was searching for the result of the bid, he would obviously not have added the concept "Qatar" to his query. Our solution framework sets out to solve problems such as this one.

Topic profiles are supposed to be expandable so that more information can be integrated when available. To specify the importance of an element in the profile, a weight is assigned to each element (see Definition 1).

Definition 1 (Topic Profile). *The profile P of a* topic *T is a set of weighted concepts t, determined by a set of strategies. Each concept has weight $w_s(t, T)$, which is determined by the strategy s, and the topic T.*

$$P_s(T) = \{(t, w_s(t, T)) | t \in s(T)\}$$

In the topic profile, the knowledge gained from named-entity recognition, tweets, and news articles will be merged with varying importance weights.

Semantic Enrichment of Topics The search topics are short and consist on average of 3.5 words. They contain abbreviations, part of names, and nicknames. One example (see Table 1) is the first name "Jintao" (in example topic Example002: "Jintao visit US") which refers to the President of the People's Republic of China. However, in tweets he may also be referred to as "President Hu", "Chinese President", etc. If these variants of a person's name and titles are considered when building a topic profile, a wider variety of tweets can be found. We utilize

[3] Language detection, `http://code.google.com/p/language-detection/`

Table 1. Example of named-entity recognition and possible concepts in the topic.

Topic (Example 002)	**Jintao** visits US	
Name of the Entity	**Annotated Text**	**Possible Concepts**
Hu Jintao	Jintao	Hu, Jintao, Hu Jintao

the well-known named-entity recognition (NER) service DBPedia Spotlight[4] to identify names and their synonyms from text snippets. A snippet can be a topic, a tweet or a Web page.

Semantic Enrichment Based on Tweets & News The information from other sources, not just the original topic string, can also be utilized to enhance the topic profile. The named entities identified in the topics can be used to search in corpora for related entities. One obvious choice is of course the Twitter corpus itself: the entities identified in the topics are used to find related tweets, they are NER processed and the additional entities are added to the profile (selective automatic query expansion). Since the provided TREC example topics are often related to current news, we also consider an external corpus of news articles as additional source of topic enrichment. We collected the titles and abstracts of news articles written in the time frame of the Twitter corpus. For each topic, we extracted the headlines and abstracts that contained entities identified in the topic. Applying NER yielded another set of new entities.

Topic Profile Aggregation Having identified additional named entities to enrich the original topic, we aggregate the entities into the final profile. Different weights are attached to different sources. For example, a greater weight may be assigned to the given topics than the additional entities extracted from tweets and related news articles. We expect such a weighting scheme to reduce the amount of query drift considerably.

Topic Profile to Query Translation Since we extracted the formal names of the entities mentioned in the topics, related tweets, and news articles, the query can be expanded to a more comprehensive version. Different strategies are used when we consider to integrate the information from these different sources.

To reduce the risk of query drift, we performed the following steps:

- Each word in the topic is assigned weight w.
- If the annotated text in the topic contains more than one word, the name of the entity is appended with weight w. Otherwise it (i) is dropped, or (ii) it is appended with weight $\frac{w}{2}$.
- Each word in the names of the extracted entities is appended with half the weight of the entities. This is useful when the name is not fully mentioned (see Table 1). For example, "Jintao" is the first name of the Chinese president, but the name of the entity is "Hu Jintao". Sometimes he is also mentioned with his last name "Hu".

[4] DBpedia Spotlight, `http://spotlight.dbpedia.org/`

After expanding the query with the entities extracted from the topic profile, we submit the query to the search system and retrieve related tweets or related news articles, which are prerequisites for the further expansion of the query.

Using Top Entities in Related Tweets. We integrate the information from related tweets in order to learn how users talk about a topic, which words they are using, which people they are referring to and so on.

The related tweets are the tweets we retrieve by searching the corpus with the query expanded with entities extracted from the topic profile. We select the top entities as the source of query enrichment. The possible concepts in them are considered to be appended to the query string. The total weight of these concepts can be adjusted. We apply similar weighting strategies as already described earlier:

- If the annotated text in the topics contains more than one word, we append the name of the entity with weight w as a whole concept. Otherwise it is appended with weight $\frac{w}{2}$.
- Append each word in the multiple-word concept as concepts and assign them half the weight of the original multiple-word concept.

Using top entities in related news articles Often, users write postings on Twitter about what is going on at the moment in the world. News articles contain details about the most important events. We expect to be able to benefit from searching for related news articles to find additional entities that can be included in the topic profiles.

The related news articles can be retrieved by searching a news corpus. We select the top entities from news articles that were published four days before and after the query time. Again, we use the same strategy to compute the weight of concepts derived from each entity.

2.3 Indri Query Language

In the last step, the topic profile is converted to the Indri query language which supports the assignment of weights to query concepts. Entities consisting of more than one concept are treated as a phrase (Indri's #1(....) operator). As an example, consider the Indri query below, which was enriched with named entities derived from news and related tweets (topic MB002 "2022 FIFA soccer"):

```
#weight( 0.21265 FIFA 0.00838 Brazil 0.00838 #1(The Bahamas)
0.00838 #1(FIFA Beach Soccer World Cup) 0.13932 #1(Association football) 0.00599 Eurosport
0.00599 #1(West Bromwich Albion FC) 0.00599 #1(2010 Commonwealth Games) 0.00539 Italy
0.00479 #1(Nassau Bahamas) 0.00329 Bahamas 0.00299 Basel 0.00269 Yahoo
0.00240 #1(Frank Purdy Lahm) 0.00240 Futsal 0.00240 #1(FC Basel)
0.00240 Sport 0.00210 Beach 0.00210 Soccer 0.01213 World 0.04015 #1(FIFA World Cup)
...
0.03333 game 0.03333 series )
```

Table 2. Statistics of the Twitter corpus, the external news sources and the extracted named entities.

Corpus	#Elements
Crawled Twitter 2011 Corpus	14,958,450
English Twitter Corpus	4,766,901
RSS News Feeds	62
News Articles	13,959
Entities extracted from Twitter Corpus	6,193,060
Entities extracted from News Articles	357,559

3 Experimental Setup

3.1 Twitter Corpus

The corpus consists of approximately 16 million tweets, posted over a period of 2 weeks (January 24 until February 8th, inclusive). Since over time, less tweets are available for public access, we were only able to crawl fifteen million tweets (crawled in June/July), of which nearly five million tweets were detected to be written in English. Employing NER on the English tweets resulted in a total over six million named entities among which we found approximately 0.14 million distinct entities.

The external news corpus was derived by extracting articles from 62 RSS feeds of prominent news media such as BBC, CNN or New York Times. A precise overview of the numbers can be found in Table 2.

3.2 Submitted Runs

We submitted four runs: basicWISTUD (automatic, no external or future evidence), manualWISTUD (manual), dbpWISTUD (automatic, external and future evidence), and mulnewWISTUD (automatic, external and future evidence). They are described in turn.

basicWISTUD A separate index was created for each topic that included all tweets up to the topic's time stamp only (to avoid dealing with corpus statistics that are computed over future tweets). Apart from filtering out non-English tweets, we also removed re-tweets, tweets with less than 100 characters, tweets with less than 50 characters if URLs are ignored and tweets with words that contain a single letter three or more times in sequence (e.g., "ooooooooh" or "aaaaaaah"). Language modeling with relevance model RM2 [4] was employed. Up to 1000 results were returned per topic. No additional processing was performed to take the time-based ordering of the task into account.

manualWISTUD The manual run was created by processing each topic for 5 minutes as follows: one of the paper's authors formulated queries, submitted

them to a MySQL engine[5] and scanned the results of the English-tweet corpus sorted in descending order of tweet time. Tweets the annotator deemed relevant were marked. Tweets being duplicates of already marked tweets were subsequently ignored. The annotator did not follow hyperlinks mentioned in tweets, only the tweet text itself was considered (though it was sometimes possible to determine the potential informativeness based on the URL itself, e.g. a link `http://www.bbc.co.uk/.../` was considered more informative than `http://bit.ly/rrxSt9`). No external source (e.g. news articles) were used to learn more about a topic, potential new query concepts were learnt while scanning the tweets. Once the 5 minutes were up, the next topic was processed. On average, 20.8 tweets were marked as relevant. The minimum was 0 (topic MB047) and the maximum was 42 (MB039). The tweet time of the oldest manually marked tweet was recorded and one more query (also created by the annotator) was submitted; all tweets retrieved this way with a time stamp older than the manually selected ones were appended until a maximum of 1000 tweets was reached.

dbpWISTUD This run integrated background knowledge from DBpedia and applied the semantic enrichment strategies described in Section 2.2. Therefore, we extracted entities from the topic description and the top related tweets by means of different NER services (DBpedia Spotlight, Alchemy API, and Zemanta). Only those entities for which we could identify a DBpedia URI were taken into account while profiling the topic. Each topic was therefore first represented via a set of weighted DBpedia concepts and then transformed into a set of weighted terms by exploiting the labeling information from DBpedia. The entire process including the weighting is described in Section 2.2. The final topic profile was constructed as an accummulation of the initial profile (inluence = 80%), which was generated based on the topic description, and the expanded profile (inluence = 20%), which was generated based on the related tweets. We then translated the topic profile into the query according to Indri's syntax and retrieved the top 100 tweets via the Inri search engine using relevance model RM2 [4]. Finally, the retrieved tweets were then ranked according to their recency. On average, 34.9 tweets were marked as relevant. The minimum was 0 (MB033) and the maximum was 122 (MB 020). Across all topics, the precision at 30 documents (P@30) is 0.301. The best four performing topics achieved $P@30 = 0.833$ (MB020, MB021, MB022 and MB030).

mulnewWISTUD Besides potentially related entities from tweets, this run also took the related news into account. The topic profiling was as same as in the dbpWISTUD run, but we issued the search query not only to the Twitter corpus but also the news articles corpus. Then, besides the entities retrieved from the tweets, we also aggregated the entities of the top ranked news articles into the final topic profile. The expanded parts for news and tweets were both weighted with 20%. On average, 30.7 tweets were marked as relevant. The minimum was

[5] Regular expressions were used, e.g. "%bbc%cut%" for searching the corpus for tweets relevant to the topic "BBC World Service staff cuts" (MB001).

Table 3. Results of the official runs (revised relevance assessments). Reported are precision at 30 tweets (P@30) and mean average precision (MAP) for the full set of 49 topics (all) and the 33 topics with highly relevant tweets (high). The last two columns show the number of topics for which the P@30/MAP performance was better than the median P@30/MAP across the runs submitted by all track participants.

Run	Topics	P@30	MAP	>Median P@30	>Median MAP
basicWISTUD	all	0.0993	0.1110	7	14
	high	0.0323	0.1207	3	11
dbpWISTUD	all	0.3014	0.2291	25	29
	high	0.1051	0.1999	16	19
mulnewWISTUD	all	0.1959	0.1553	14	18
	high	0.0859	0.1508	14	16
manualWISTUD	all	0.3946	0.2719	42	40
	high	0.1242	0.2705	22	24

0 (MB016, MB033 and MB048) and the maximum was 123 (MB020). Here, $P@30 = 0.196$ across all topics. The two top performing topics achieved $P@30 = 0.833$ (MB022 and MB030).

4 Results

The results of our official runs are reported in Table 3. The evaluation measures are P@30 and mean average precision (MAP). The results for the original topic set, consisting of 49 topics[6] are shown in rows marked with *all*, while the subset of 33 topics that contain highly relevant tweets are shown in rows marked *high*. Shown are the results of the revised relevance assessment process (v2.0). The final two columns list the number of topics for which the P@30/MAP performance of our runs improved over the median P@30/MAP derived from all runs submitted by the participants of this task.

The manual run outperforms the automatic runs on all measures by a considerable margin. This is not surprising, as the manual run involved a great amount of human effort. In 42 of the 49 all relevant topics, the manual result outperforms the median P@30 across all submitted runs. The fraction of topics that improve over the median decreases when considering only the topic set with highly relevant tweets, a result which can be explained by the fact that the human annotator only considered the tweet text and possibly the URL string, but did not follow the hyperlinks present in the tweets. As will be described below, many relevant tweets do contain URLs. Among the automatic runs, the topic enrichment process with DBPedia Spotlight performs best, with $P@30 = 0.3$ (all) and $P@30 = 0.1$ (high). Extracting related entities from news articles has a less positive effect which may be caused by query drift.

[6] One topic was dropped from the evaluation.

Relevance Judgments Analysis Our initial analysis of the relevance judgments[7] revealed a number of interesting characteristics. The average length (number of characters) of the relevant tweets is 109.4, the minimum being 16. Thus, by filtering out short tweets, we are, inadvertently, already removing a number of relevant tweets in the pre-processing step. As already stated, in neither our manual nor our automatic runs were tweeted hyperlinks followed. This is a necessary step however, as 81.9% of the relevant tweets contain URLs. When only considering the highly relevant tweets this percentage rises further: 95.3% of them contain URLs. In both judgment sets - all relevant and highly relevant only - the minimum length of extra information besides the URL was found to be 0 (the average being 87.0 for all relevant and 85.34 for the highly relevant judgments), which means that some relevant tweets actually contain only a URL and no further information.

The DBPedia entity extraction was found to perform well, among the relevant tweets, in 86.8% (86.0% for the highly relevant) one or more named entities could be extracted. On average, relevant tweets contain 2.8 entities.

Finally, the language detection process we performed as a pre-processing step also removed a small number of tweets from the pool of relevant tweets: 94.8% of the relevant tweets (95.5% of the highly relevant tweets) were detected to be in English; tweets consisting only of a URL were not considered to be in English.

5 Discussion

Three topics for which our runs *dbpWISTUD* and *mulnewWISTUD* performed worse than the median are: MB 001[8], MB 003[9], and MB 043[10]. For these topics, Table 4 shows the top entities and their assigned weights. After an initial investigation, we found a number of reasons for the poor effectiveness of our system in these case:

- The NER services sometimes fail. For example, the word "staff" in MB 001 was recognized as an entity which is related to a class of stick-shaped objects. The word "olive" in MB 043 was annotated with an entity which was linked to a music band. One solution to this problem is to limit the types of entities that we adopt while employing NER on topics. The context of the topics should also be considered in order to filter out those obviously non-relevant entities.
- The multiple-word entities received weights which were disproportionate: other useful information was not appropriately weighted.

6 Conclusion and Future Work

In this paper, we have presented the search framework we created for the Microblog track 2011. We investigated the suitability of semantic enrichment tech-

[7] Note, that this analysis was conducted on the initially released relevance assessments.
[8] BBC World Service Staff cuts
[9] Haiti Aristide return
[10] Kucinich olive pit lawsuit

Table 4. Receognized entities and their assigned weights for three example topics.

Topic	**MB001**	**MB003**	**MB043**
Entity and Weight	BBC World Service	Haiti	Dennis Kucinich
	0.30124	0.37274	0.28660
	BBC	Jean Bertrand Aristide	Olive_(band)
	0.20963	0.13277	0.18750
	Staff_(stick)	Return Transnistria	Kucinich
	0.07143	0.10000	0.11852
	Cuts	Jean	Dennis
	0.07143	0.06212	0.11852
	World	Bertrand	band
	0.03960	0.05819	0.09375

niques to improve the retrieval effectiveness. The fact that our run, enriched with DBPedia knowledge, performs better than the median across all submitted runs for half of the topics, leads us to the conclusion that semantic enrichment is a technique which should be investigated further in this setting. Future work will focus on the extraction of content from hyperlinks present in tweets, as our analysis has shown that the majority of relevant tweets contain URLs.

Acknowledgements We would like to thank Wen Li from the Multimedia Information Retrieval Lab at TU Delft who kindly helped us with crawling the Twitter corpus. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no ICT 257831 (ImREAL project[11]).

References

1. van der Zee, B.: Twitter triumphs. Index on Censorship **38**(4) (2009) 97–102
2. Abel, F., Gao, Q., Houben, G.J., Tao, K.: Analyzing User Modeling on Twitter for Personalized News Recommendations. In Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N., eds.: Proceedings of the 19th International Conference on User Modeling, Adaption and Personalization (UMAP), Girona, Spain. Volume 6787 of Lecture Notes in Computer Science., Girona, Spain, Springer (July 2011) 1–12
3. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Symposium On Document Analysis and Information Retrieval. (1994) 161–175
4. Lavrenko, V., Croft, W.B.: Relevance based language models. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '01 (2001) 120–127

[11] `http://imreal-project.eu`