Automatic Query Expansion Based on Tag Recommendation [*]

Vitor Oliveira[1,2] Guilherme Gomes[1] Fabiano Belém[1] Wladmir Brandão[1]
Jussara Almeida[1] Nivio Ziviani[1,2] Marcos Gonçalves[1]

[1]Universidade Federal de Minas Gerais, Brazil
{vitorco, gcm.gomes, fmuniz, wladmir, jussara,
nivio, mgoncalv}@dcc.ufmg.br

[2]Zunnit Technologies, Brazil
{vitor, nivio}@zunnit.com

ABSTRACT

We here propose a new method for expanding entity related queries that automatically filters, weights and ranks candidate expasion terms extracted from Wikipedia articles related to the original query. Our method is based on state-of-the-art tag recommendation methods that exploit heuristic metrics to estimate the descriptive capacity of a given term. Originally proposed for the context of tags, we here apply these recommendation methods to weight and rank terms extracted from multiple fields of Wikipedia articles according to their relevance for the article. We evaluate our method comparing it against three state-of-the-art baselines in three collections. Our results indicate that our method outperforms all baselines in all collections, **with relative gains in MAP of up to 14% against the best ones.**

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query formulation

General Terms

Algorithms, Experimentation

Keywords

Query Expansion, Tag Recommendation, Wikipedia

1. INTRODUCTION

Ranking of information items based on their estimated relevance for an information need expressed as a search query is the key technology of modern search engines. However, expressing such information need with a single query is not an easy task. These queries are often ambiguous or vague. Query expansion by means of direct or indirect feedback from the user is one way of solving this problem.

Pseudo-relevance feedback (PRF) [5] is one of the most used methods for automatic query expansion. PRF uses the top ranked documents retrieved by the original query, which are assumed to contain relevant terms that help to better express the user's information need, as source of new terms for the query expansion. However, the top retrieved documents may not provide good terms for expansion, particularly for "difficult queries" with few relevant documents in the collection which do not share many relevant terms. In such cases, PRF may negatively impact the results. In contrast, the use of external sources of information about query concepts (e.g., thesauri or information available on the Web) for automatic query expension has been exploited in the past, resurging with strength recently.

Once such source is Wikipedia. Indeed, Wikipedia has been exploited for automatic query expansion, mainly for queries related to specific entities or narrow and defined concepts [9]. In general, a defined entity is described in Wikipedia with an article with information and references of relevance. For the task of query expansion, entity related queries can be associated with one or more related articles in Wikipedia, which act as source of terms for expansion. One issue that must be addressed is how to select the best terms for query expansion from a Wikipedia article, considering that some of these articles are very lengthy and structured in complex ways (with abstract, sections, infoboxes, etc).

Another possible source of terms for automatic query expansion is collaborative social annotation (i.e., tags). Indeed, recent studies have shown that tags have great potential to improve automatic object classification [3] and searching [6]. Recently, tags have also been used as a new source of terms for query expansion [7]. A potential problem with such use is that tags may contain a lot of noise (e.g., misspellings). Moreover, users create and assign tags with several purposes other than only describe an entity/concept. For instance, they may use tags for personal organization (e.g., 'toread') or for expressing opinions (e.g., 'cool' or 'dislike'). Thus, many terms that are *not* related to the query concepts may be extracted and used for query expansion, which ultimately may hurt search effectiveness.

We here propose to combine, expand and improve several of the aforementioned ideas into a new method for expanding entity (or narrow concepts) related queries, which automatically filters, weights and ranks terms extracted from Wikipedia articles related to the original query. Our method

[*]This work is supported by the INWeb (grant 57.3871/2008-6) and by the authors grants from CNPq and FAPEMIG.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.

is based on state-of-the-art tag recommendation methods that exploit both heuristic metrics to estimate the descriptive capacity of a term and the Wikipedia page structure [1]. Originally proposed for the context of tags, we here apply these recommendation methods to weight and rank terms extracted from multiple fields of the Wikipedia article (e.g., title, abstract, sections) according to their estimated relevance for the article. The hypothesis is that the most relevant terms for the article, according to the tag recommendation method, are also the most relevant ones for the query expansion, since article and query are semantically related.

We compare our proposed method against a state-of-the-art social annotation method [7], a state-of-the-art method that also exploits Wikipedia [9], and a Language Model PRF method [5] on three collections. The social annotation method exploits user annotated objects as source of terms for query expansion, whereas the PRF method is the Indri's implementation of Lavrenko's relevance model [5]. The Indri search engine was used as information retrieval system in all experiments. We find that our method outperforms all baselines in all three collections, with relative gains, in terms of Mean Average Precision (MAP), of more than 23% over the original queries and **up to 15% over the best baseline method.**. Moreover, these results are close to an ideal oracle that always knows whether to expand or not.

2. RELATED WORK

Pseudo-relevance feedback (PRF)[5, 10] has been the most used method for automatic query expansion. We here use Lavrenko's relevance model, a *de facto* PRF standard, implemented in the Indri system as one of our baselines.

But there have been also proposals to exploit external sources of information for that purpose, with the motivation that query expansion may fail due to the lack of relevant documents in the local collection. Cui et al. [2] proposed a query expansion approach based on mining user logs, which exploits the correlation between query terms and document terms to select expansion terms for new queries. Xu et al. [9] proposed a query dependent PRF method that uses Wikipedia as a resource. In general, the pages collected from Wikipedia are used as a set of pseudo-feedback relevant documents tailored to the specific query.

The closest approach to ours takes advantage of Web 2.0 social annotations (tags) for query expansion [7]. It uses a ranking approach to select terms for query expansion. First, a co-occurrence based method is used to choose a number of candidate terms for expansion. These terms are extracted from various fields (including tags) of pages collected from Delicious[1]. Candidate terms that most often co-occur with terms in the query are considered as the most likely for using in the expansion. Next, it runs expanded queries using each candidate term and the original query to check whether it improves the average precision of the results. If so, the candidate term is considered relevant for expansion.

3. QUERY EXPANSION BASED ON TAG RECOMMENDATION

This work is inspired by a recently proposed method for recommending relevant tags for a target Web 2.0 object o which jointly exploits 3 dimensions: (i) term co-occurrences with tags previously assigned to o, (ii) terms extracted from multiple textual features (e.g., title, description)of o, and (iii) metrics of tag relevance, particularly heuristic metrics that capture the capacity of a candidate term to describe o's content [1]. We here apply this tag recommendation method to select terms for expanding a given query, focusing particularly on dimensions (ii) and (iii). Specifically, we apply it to filter terms from multiple fields (title, abstract, article sections, references) of a Wikipedia article semantically related to the original query (dimension (ii)), and use the proposed heuristic metrics (dimension (iii)) to rank the filtered terms according to their relevance for the Wikipedia article. The assumption is that, given that Wikipedia article and query are semantically related, the terms that are more relevant to the article are also more relevant for the query expansion task. Currently, we do not exploit term co-occurrences with tags (dimension (i)), as tags are absent in Wikipedia.

3.1 Tag Recommendation

The tag recommendation methods proposed in [1] exploit various textual features commonly associated with Web 2.0 objects. These features comprise self-contained blocks of text, usually with a well defined functionality, such as title and description [3]. Generally speaking, the tag recommendation problem can be defined as [1]: given a set of textual features $F_o = \{f_o^1, f_o^2, ..., f_o^n\}$, associated with an object o, where each element f_o^i is the set of terms in textual feature i associated with o, generate a set of candidate tags C_o and recommend the k most relevant tags in C_o.

In [1], the authors generate candidate tags by exploiting co-occurrence patterns with tags previously assigned to the target object and extracting terms from other textual features (notably title and description) associated with it. They then estimate the relevance of each candidate by applying various quality metrics. The metrics that led to the best recommendation results, outperforming various previous methods, are heuristics that try to capture the capacity of a term to describe the object's content. These descriptive metrics are: Term Spread (TS), Term Frequency (TF), weighted Term Spread (wTS) and weighted Term Frequency (wTF).

The Term Spread of a candidate term c (we disregard stopwords) for a target object o, $TS(c, o)$, is defined as the number of textual features of o that contain c. TS exploits the structure of the object, an aspect that has only very recently been applied to tag recommendation [1], being very different in nature from, for example, the more traditional inverse document frequency (IDF) that captures the overall frequency of a term in a collection. TS assumes that the larger the number of features of o containing c, the more related c is to o's content. The maximum TS value is given by the number of textual features considered. In contrast, the Term Frequency of candidate c in a object o, $TF(c, o)$, is defined by considering all textual features associated with o as a single list of terms, and simply counting the number of occurrences of c in it. Thus, unlike TS, TF does not exploit the structure of the object in terms of textual features.

Both TS and TF assume that each textual feature has the same descriptive capacity, which may not be true (e.g., the title may carry terms that more accurately describe the object's content than the comments). Towards capturing the importance of each feature, Belém *et al.* [1] proposed weighted versions of TS and TF, namely weighted Term Spread (wTS) and weighted Term Frequency (wTF), which

[1] http://delicious.com

weight each term based on the average descriptive capacities of the textual features in which it appears. The authors estimate the descriptive capacity of a textual feature by the Average Feature Spread (AFS) heuristic [3], Let the Feature Instance Spread of a feature $\mathcal{F}_o^i$ associated with object o, $FIS(\mathcal{F}_o^i)$, be the average TS over all terms in $\mathcal{F}_o^i$. $AFS(\mathcal{F}^i)$ is defined as the average $FIS(\mathcal{F}_o^i)$ over all instances of $\mathcal{F}^i$ associated with objects in a training set $\mathcal{D}$. The wTS and wTF metrics are then defined as:

$$wTS(c,o) = \sum_{\mathcal{F}_o^i \in \mathcal{F}_o} j, \text{ where } j = \begin{cases} AFS(\mathcal{F}^i) & \text{if } c \in \mathcal{F}_o^i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$wTF(c,o) = \sum_{\mathcal{F}_o^i \in \mathcal{F}_o} tf(c, \mathcal{F}_o^i) \times AFS(\mathcal{F}^i) \tag{2}$$

3.2 Query Expansion

In order to expand a given query, we first identify an external source of information related to the query. We here focus on Wikipedia articles as the primary source. Given a query and its related Wikipedia article, our goal is to produce a ranked set of terms that can be used for expanding the query. To that end, we apply the tag recommendation method described in Section 3.1. Specifically, the method is applied to the given Wikipedia article w (i.e., object), taking w's title, summary (first section), largest section (other than the summary) and text of references as the set of textual features F_w of w. We build a set of candidate terms C_w by extracting them from these textual features, and we use one of the descriptive heuristic metrics TS, TF, wTS or wTF to rank these terms according to their relevance to w.

To build each expanded query, we use the #weight belief operator of the Indri system [8], as it allows more control on the impact of each term on the query for obtaining the final score. The expanded query is formed as $\#weight(\delta_{fb} \times Q_{ori}\ (1 - \delta_{fb}) \times Q_{exp})$, where Q_{ori} is the original query, Q_{exp} corresponds to the expanded query, and δ_{fb} defines the relative importance of each component.

We evaluate two types of expansion: (1) the Q_{exp} component is composed of only expansion terms, with no weights, and (2) the Q_{exp} component is composed of the terms with associated weights, where the weight of a term is given by one of the heuristic metrics, namely TF, TS, wTF and wTS.

Wikipedia is currently the primary source of information for our method, although other Web 2.0 sources with multiple textual features (e.g., LastFM) can be used in the future for specific queries (e.g., celebrities, artists). For each test collection, we *automatically* select those containing a clear and specific entry on Wikipedia using the query classification method proposed in [9]. In most cases, the selected queries contain entities (e.g., a person or place) or a specific narrow concept (e.g., "dinosaurs"), being referred to entity queries. Next, we describe how we extract the textual features from the Wikipedia entries corresponding to the selected queries as well as the coverage of the entities in the used collections.

4. EVALUATION METHODOLOGY

4.1 Collections

We use three TREC collections for evaluating web retrieval quality, namely ClueWeb09 Category B (or simply ClueWeb09B), WT10g and GOV2. Table 1 summarizes these collections, presenting the number of queries, the number of entity queries identified with a corresponding Wikipedia page, the number of documents and average query length. For each entity query[2], the corresponding Wikipedia article associated with it was processed, and a list of candidate terms with their respective metrics were generated. We note that, in our evaluation, we used *all* queries in each collection with our methods and the baselines. We discuss how we handled non-entity queries that are not associated with Wikipedia articles in Section 4.3.

	ClueWeb09B	WT10g	GOV2
Total # of queries	98	100	149
# of entity queries	40	61	120
# docs	50,220,423	1,692,096	25,205,179
Avg. query length	1.96	2.31	2.59

Table 1: Summary of the Collections

For our experiments we used the title, summary, the body of the article (content of the main sections), and the references of the Wikipedia articles as our textual features for term extraction and metric computation. These fields were *automatically* extracted with a special purpose parser designed by us for extracting them. Thus, our method is completely automatic, from the identification of the Wikipedia page up to the extraction of the required fields.

4.2 Baselines

We selected three baseline methods for comparing against our solution. The first method is a *de facto* standard pseudo-relevance feedback model, here referred to as PRF, which is based on the Indri's implementation of the Lavrenko's relevance model [5]. In PRF, a set of N documents is retrieved and used to form the expanded query Q_{exp} by adding the top k most likely terms using the Indri's $\#weight$ operator.

The second baseline corresponds to our own implementation of the method proposed in [9], which also exploits Wikipedia as a repository of entities and uses a pseudo-relevance feedback framework for query reformulation. The method first identifies the most representative entity in Wikipedia for a given query, and, if it exists, ranks the terms extracted from the identified Wikipedia page by their TF-IDF values, selecting the top k terms for expansion. We refer to this baseline as WE, which stands for Wikipedia entities.

The last baseline is also our own implementation of a recently proposed method that exploits social annotation systems (e.g., Delicious) as a source of terms for query expansion [7]. Thus, we here refer to it as Social Annotation-based query expansion or SA. The SA method first selects and extracts candidate terms from Delicious, and then ranks them based on a measure of term importance for the query and the learning-to-rank RankSVM technique. It was shown to outperform PRF techniques based on experiments in three TREC collections and in a sample of Delicious [7].

4.3 Setup and Evaluation Metrics

We used the Indri 2.6 search engine [8] as our basic retrieval system. Retrieval effectiveness is measured in terms of Mean Average Precision (MAP) for the top 1,000 doc-

[2] The actual set of entity queries and corresponding Wikipedia pages used in our experiments can be found in
http://dl.dropbox.com/u/84084/GOV2.txt,
http://dl.dropbox.com/u/84084/clueweb09.txt,
http://dl.dropbox.com/u/84084/WT10g.txt.

Method	Unweighted Query Expansion			Weighted Query Expansion		
	ClueWeb09B	WT10g	GOV2	ClueWeb09B	WT10g	GOV2
Orig. Query	0.141	0.195	0.294	0.141	0.195	0.294
PRF	0.141	0.202	0.315	0.141	0.202	0.315
SA	0.142 (0.142)	0.194 (0.199)	0.300 (0.317)	0.143 (0.142)	0.190 (0.195)	0.296 (0.313)
WE	**0.162 (0.165)**	0.183 (0.187)	0.276 (0.282)	**0.159 (0.161)**	0.182 (0.186)	0.275 (0.282)
TF	**0.170 (0.174)**	0.219 (**0.222**)	0.324 (**0.331**)	**0.168 (0.172)**	**0.225 (0.229)**	0.324 (**0.331**)
TS	**0.168 (0.172)**	0.211 (0.215)	0.318 (0.325)	**0.166 (0.170)**	0.212 (0.216)	0.319 (0.326)
wTF	**0.169 (0.173)**	**0.221 (0.225)**	0.324 (**0.331**)	**0.167 (0.171)**	**0.225 (0.230)**	0.323 (**0.331**)
wTS	**0.167 (0.171)**	0.212 (0.216)	0.316 (0.323)	**0.168 (0.172)**	0.213 (0.217)	0.318 (0.325)

Table 2: Map Results for Weighted and Unweighted Query Expansion. Best results, including statistical ties with 95% confidence level, are shown in bold for each scenario.

uments. We report MAP results for all (entity and non-entity) queries, before and after expansion. When selecting terms, we remove stopwords but do not perform stemming.

For all baselines and our methods, only the top 50 candidate terms according to their respective selection methods and weighting functions are selected for expansion. For the PRF method, we fixed parameters N and k equal to 10 and 50, and set the $\#weight$ operator parameter δ_{fb} equal to 0.5, as these values provided the best results in previous work [7]. For the SA baseline, we used the same parameter values used in the original work. Moreover, since the SA baseline is the only supervised one, we used a leave-one-out procedure for training, i.e., we selected one query at time for testing (i.e., for ranking its potential terms) using all other queries as training[3]. Tuning of the RankSVM algorithm was performed using cross-validation in the training set.

For our proposed methods and for the WE baseline, we used two strategies to deal with non-entity queries[4]: (1) we kept the original query as it is, and (2) we used the PRF method. Note that in the original proposal of the WE baseline, the authors also treated ambiguous queries which could be associated with many entities (Wikipedia pages). As their proposed disambiguation method demonstrated to have a rather low effectiveness (in the authors' own words) and did not produce consistent gains for this type of query, we did not consider such ambiguous queries as entity queries in order to apply our method and the WE baseline. We leave the task of disambiguating these queries for the future. Nevertheless, we would like to stress that **all** queries of the three collections were used by all reported methods (ours and the baselines). Finally, similar strategies were applied to the SA baseline for queries for which it was not possible to associate Delicious tags that co-occur with the query terms.

5. EXPERIMENTAL RESULTS

In this section, we report MAP results for all analyzed methods, comparing them using statistical tests (i.e., two-tailed paired Student's t tests) with 95% confidence level. Specifically, we performed a pairwise comparison of all methods, applying a paired difference test for each pair of methods on each query to test whether their average results are statistically different with 95% confidence. To improve readability, Table 2 presents only MAP results, showing in bold the best results for each analyzed scenario (i.e., unweighted and weighted query expansion), along with statistical ties.

The table shows the results for each baseline as well as for our strategy applied with each considered descriptive heuristic - TF, TS, wTF and wTS - used as term ranking criterion and, for weighted expansion, weight factor. For PRF, instead, all reported results refer to the method implementation available in the Indri engine, which does apply weights to terms. Moreover, for our approach as well as for the WE baseline, we report two sets of results for each scenario: one obtained when no expansion is performed for non-entity queries, and the other (in parenthesis) obtained when pseudo-relevance feedback (PRF) is applied to them. Similar results are presented for the SA baseline for cases in which there were no Delicious tags cooccurring with the query terms (i.e, no expansion and expansion with PRF). In the following discussion, we refer to these two variations of each method as the method *with* and *without PRF*.

5.1 Unweighted Query Expansion

Starting with the results for unweighted query expansion, Table 2 shows that the PRF method significantly enhances the retrieval performance over the original query only in the GOV2 collection (gains of 7%). The same is true for the SA baseline with PRF applied to non-entity queries. The lack of improvement of the SA baseline *without* PRF on both WT10g and GOV2 may be explained by the low frequency of co-occurrences between queries in these two collections and tags in our Delicious dataset. In such cases, the use of PRF on non-entity queries helps the performance of SA.

The WE baseline produced reasonable gains (17%) in the ClueWeb09B. However, in the other two collections, the method led to MAP degradation when compared to the original queries. A deeper investigation of the expansions performed by this method in these collections revealed that, for some entity queries, information automatically extracted from links and anchor texts included rare terms not associated with the query as well as terms from Wikipedia in other languages. These terms were promoted and selected due to the use of the IDF metric by this baseline, ultimately hurting the performance of several queries. This behavior was not reported in the original work [9], although the authors explicitly indicated that they used link information. We hypothesize that the authors may have performed some type of filtering, although this is not described in the paper. The better behavior of the WE baseline in the ClueWeb09B collection may be explained by the following observations: 1) this collection is more recent and perhaps more aligned with the also recent Wikipedia pages used; and 2) this collection is larger and less focused, better capturing the characteristics of the Web. For the other two collections, in contrast, there may have been some vocabulary mismatch.

On the other hand, our new approach with most of the

[3] Notice that we basically used almost the entire query set for training, which in fact may have helped this method.

[4] It is also important to stress that the same set of Wikipedia pages was used for our method and the WE baseline.

proposed metrics (mainly TF and wTF) produces large performance gains over the original query and over all baselines in all three collections, but WE in ClueWeb09B, where there is a statistical tie. Moreover, there is a trend towards some gains from jointly using our method with PRF for non-entity queries over not using PRF: for given metric and collection, the results obtained with and without PRF are statistically different in some cases (e.g., for wTF in GOV2).

When comparing the results produced by our method, we find that there is no clear winner among the four metrics in ClueWeb as they lead to statistically tied MAP results. However, there is a slight tendency for a superior performance, on average, of wTF in WT10g and GOV2, mainly when used together with PRF. Overall, these results indicate the capacity of all metrics of extracting good and relevant terms from the Wikipedia articles for entity queries, which, in turn, seems to be an excellent source for expansion terms.

In terms of quantitative gains, our best results outperform the MAP of the original queries by up to 23%, 15%, and 12% in ClueWeb09B, WT10g, and GOV2, respectively. When the comparison is against the best baseline in each collection, i.e., WE in ClueWeb09B and PRF in GOV2 and in WT10g, there is a statistical tie in ClueWeb09B and gains in the order of 14%, and 5% in the last two collections.

5.2 Weighted Query Expansion

We now turn our attention to the effectiveness of applying weights to terms in the query expansion process (weighted query expansion), as performed by the baselines as well as by our proposed method. Note that, in our experiments, weights did not have impact on the performance of the SA baseline in any collection. These results are in contrast to those reported in the original work [7], in which the authors reported a 13% improvement over the version without weights in one of three collections. This different behavior may be due to the use of a different learner: we here use rankSVM, instead of the private and publicly unavailable ListNet. However, our results are not completely inconsistent with the original work, as both studies found a statistical tie between both unweighted and weighted versions of SA in (at least) two of the three analyzed collections. Moreover, we note that even if we inflate the SA results by 13% (the maximum improvement reported in [7]), this would not be enough to outperform our results. Weights also do not greatly impact the performance of the WE baseline: although there is a slight tendency to degradation in effectiveness, those are not statistically significant.

In general, we find that, like with the unweighted version, our approach with any considered metrics used as both ranking criterion and weight factor produces significant MAP gains over all baselines in all collections, but WE in ClueWeb09B. Most of the gains are similar to those presented in the previous section, as, in various cases, results with and without weights are statistically tied, although there is a slight tendency of improvements with weights, particularly in WT10g. Similarly, the results tend to be superior with the use of PRF for non-entity queries over not using it, particularly in GOV2. Moreover, wTF is consistently among the best metrics in all collections, in both unweighted and weighted scenarios. Recall that wTF is one of the most complete metrics that exploit both term descriptive capacity and the structure of the page into multiple textual fields.

In sum, our proposed method, mainly wTF, can be used as an effective strategy for capturing relevant descriptive terms given a set of textual features. Moreover, since the weights are not expensive to compute, and their use does not hurt performance and have a slight tendency to help, we suggest to use them in most cases. Finally, we also advocate for the use of PRF for non-entity queries as it produces results that are at least as good as (and better, in some cases) than the alternative of keeping the original query with no expansion.

As a final analysis, we now compare our best approach, i.e., weighted query expansion with wTF and PRF for non-entity queries, against an ideal method that always knows whether to expand or not. To that end, we built an *Oracle* by choosing, for every single query, the best average precision value among: expansion with our best method, expansion with PRF, and no expansion at all[5]. The MAP results achieved with the *Oracle* are 0.187, 0.252 and 0.364 for ClueWeb09B, WT10g and GOV2, respectively. Comparing them against the results of our best method, we find that the latter are only around 9% worse than the *Oracle* in all collections. Thus, even if some queries, after expansion, have performance losses, very big losses are not often or they are compensated, *on average*, by large gains for other queries. *On average*, the expansion brings benefits particularly if wTF is used both as ranking criterion and weight factor for entity queries along with PRF for non-entity queries.

6. CONCLUSIONS

We combined a good source of external information (Wikipedia) and an unsupervised state-of-the-art tag recommendation method to produce a method to filter and rank terms for query expansion. Our experimental evaluation in three collections, comparing our approach against three state-of-the-art baselines, showed that the best results are obtained when the queries are expanded with the terms suggested by the wTF metric along with the respective weights and PRF is applied to non-entity queries.

7. REFERENCES

[1] F. Belém, E. Martins, T. Pontes, J. Almeida, and M. Gonçalves. Associative tag recommendation exploiting multiple textual features. In *SIGIR*, 2011.

[2] H. Cui, J.R. Wen, J.Y. Nie, and W.Y. Ma. Query expansion by mining user logs. *IEEE Transactions on Knowledge and Data Engineering*, 2003.

[3] F. Figueiredo, H. Pinto, F. Belém, J. Almeida, M. Gonçalves, D. Fernandes, and E. Moura. Assessing the quality of textual features in social media. *IP&M*, 2011.

[4] J. Giles. Special Report: Internet Encyclopedias Go Head to Head. *Nature*, 2005.

[5] V. Lavrenko and W.B. Croft. Relevance based language models. In *SIGIR*, 2001.

[6] X. Li, L. Guo, and Y. E. Zhao. Tag-based Social Interest Discovery. In *WWW*, 2008.

[7] Y. Lin, H. Lin, S. Jin, and Z. Ye. Social annotation in query expansion: a machine learning approach. In *SIGIR*, 2011.

[8] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. *In Intl. Conference on Intelligence Analysis*, 2004.

[9] Y. Xu, G.J.F. Jones, and B. Wang. Query dependent pseudo-relevance feedback based on wikipedia. In *SIGIR*, 2009.

[10] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM*, 2001.

[5] For non-entity queries, we chose the best between PRF and the original query.