AUTOMATIC QUERY EXPANSION BASED ON TAG RECOMMENDATION

VITOR CAMPOS DE OLIVEIRA

AUTOMATIC QUERY EXPANSION BASED ON TAG RECOMMENDATION

Dissertação apresentada ao Programa de Pós-Graduação em Ciencia da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciencia da Computação.

ORIENTADOR: MARCOS ANDRÉ GONÇALVES

Belo Horizonte

Agosto de 2013

VITOR CAMPOS DE OLIVEIRA

AUTOMATIC QUERY EXPANSION BASED ON TAG RECOMMENDATION

Dissertation presented to the Graduate Program in Ciencia da Computação of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Master in Ciencia da Computação.

Advisor: Marcos André Gonçalves

Belo Horizonte

August 2013

Ficha catalográfica elaborada pela Biblioteca do ICEx - UFMG

Oliveira, Vitor Campos de.

O48a Automatic query expansion based on tag recommendation / Vitor Campos de Oliveira. — Belo Horizonte, 2013.
xx, 42f. : il. ; 29cm.

Dissertação (Mestrado) - Universidade Federal de Minas Gerais – Departamento de Ciência da Computação

Orientador: Marcos André Gonçalves

1. Computação - Teses. 2. Banco de dados – Busca – Teses. 3. Sistemas de recomendação – .Teses. I. Orientador. II. Título.

519.6*73(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Automatic Query expansion based on tag recommendation

VITOR CAMPOS DE OLIVEIRA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. MARCOS ANDRÉ GONÇALVES - Orientador
Departamento de Ciência da Computação - UFMG

PROFA. JUSSARA MARQUES DE ALMEIDA GONÇALVES
Departamento de Ciência da Computação - UFMG

PROF. NIVIO ZIVIANI
Departamento de Ciência da Computação - UFMG

PROFA. VIVIANE PEREIRA MOREIRA
Departamento de Informática Aplicada - UFRGS

Belo Horizonte, 27 de agosto de 2013.

Acknowledgments

A Deus, o que seria de mim sem ele.

Aos meus pais, que sempre, incondicionalmente, me orientaram e apoiaram em todas as minhas decisões.

Às minhas irmãs, Clarissa e Julia, pela companhia e momentos de diversão. À Erika e ao Marcos pela amizade e apoio.

Ao Toninho e Helenice pelo convívio e amizade.

Aos meus novos amigos (em especial Cristiano, Marlos, TCardoso, TSalles), pelo incentivo, apoio e excelentes discussões.

Aos meus eternos amigos (Rafael, Thiago e Marcelo) pela presença (mesmo que Virtual) e amizade constante.

Aos meus professores, Marcos, Jussara e Nivio, pela orientação e experiência compartilhada.

E a Isabela, pela paciência, pelo incentivo, pela força e principalmente pelo carinho.

Resumo

Expansão de consulta é o processo de adição de novos termos às consultas feitas pelos usuários com o objetivo de melhorar os resultados de busca. Isto é especialmente útil quando a consulta original não é capaz de univocamente expressar a necessidade de informação do usuário. Tradicionalmente, os métodos de expansão de consultas são baseados em *pseudo-relevance feedback*. Os termos de expansão são selecionados entre os melhores documentos recuperados como resultado da consulta original, assumindo-se que estes documentos contêm termos úteis para a expansão da consulta. Entretanto, tal premissa pode nem sempre ser verdadeira, especialmente para consultas consideradas "difíceis". Recentemente, algumas técnicas de expansão de consultas começaram a explorar fontes externas de informação encontradas na Web, como enciclopédias on-line e sistemas colaborativos de anotação, com o objetivo de selecionar termos semanticamente ricos. Neste trabalho nós apresentamos um novo método para expansão de consultas relacionadas a entidades (ou conceitos bem definidos). Mais especificamente, nós apresentamos um método que automaticamente filtra, pondera e ordena termos extraídos de artigos da Wikipedia relacionados às consultas originais (submetidas pelos usuários). Nossa técnica é baseada em métodos do estado-da-arte para recomendação de tags que exploram métricas baseadas em heurísticas para estimar a capacidade descritiva de um dado termo. Originalmente proposto no contexto de tags, nós aplicamos estes métodos de recomendação para ponderar e ordenar termos extraídos de múltiplos campos de artigos da Wikipedia conforme a relevância deles para o artigo. Nós avaliamos nosso método comparando-o com três técnicas do estado-da-arte em três coleções. Nossos resultados indicam que o método proposto supera todas as alternativas seleciondas em todas as coleções endereçadas, com ganhos relativos em precisão média (Mean Average Precision) de 14% sobre a melhor das técnicas alternativas.

Palavras-chave: Query Expansion, Tag Recommendation.

Abstract

Query expansion is the process of adding new terms to queries posed by users in an attempt to improve search results. This is specially useful when the original query is not able to unequivocally express the user information need. Traditional query expansion approaches are based on pseudo-relevance feedback. They select expansion terms from highly ranked documents retrieved as result of the original query, assuming that these documents contain useful terms for query expansion. However, this might not always be the case, especially for queries considered "difficult". Recently, some query expansion techniques exploit external sources of information found on the Web, such as on-line encyclopedias and collaborative social annotation systems, to select semantically rich expansion terms. We here propose a new method for expanding queries related to entities (or narrow concepts). More specifically, we propose a method for automatically filtering, weighting and ranking 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 proposed method outperforms all baselines in all collections, with relative gains in MAP (Mean Average Precision) of up to 14% over the best baseline.

Keywords: Query Expansion, Tag Recommendation.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Ranking is paramount in many information services such as searching and recommendations. Particularly, ranking of information items based on the estimated relevance for an information need expressed as a search query is the key technology of modern search engines. However, expressing such an information need with a single query usually composed of a few keywords (on average two to three keywords [Baeza-Yates and Ribeiro-Neto, 2011]) is not an easy task. Moreover, in some cases these queries are very ambiguous or vague.

Query expansion by means of direct or indirect feedback from the user is one way of solving this problem[1]. The objective of query expansion is to add useful terms to the original query to facilitate the task of discriminating relevant documents from irrelevant ones.

Pseudo-relevance feedback (PRF) [Lavrenko and Croft, 2001] is one of the most used methods for automatic query expansion. Generally speaking, PRF uses the top ranked documents retrieved by the original query as source of new terms for the query expansion. Thus, it does not rely on user feedback, which, in many cases, is very hard to obtain. The assumption behind PRF is that the top ranked documents contain relevant terms that will help to better express the user's information need. However, sometimes this strategy may not work properly: 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 [Yom-Tov et al., 2005]. In such scenarios, PRF indeed may negatively impact the results. More generally, it is not clear whether query expansion using top ranked documents will always lead to improvements in search effectiveness. Thus, one challenge that has

[1]The problem of multiple concepts/meanings, on the other hand, may be solved with techniques of result diversification such as [Santos et al., 2011]).

to be faced by automatic query expansion techniques is to decide both "when and how to expand" based on the potential of improvement.

One trend that has been exploited in the past for automatic query expansion (see for instance [Rijsbergen, 1979]) but that has resurged with strength in the last few years is to make use of external sources of information about the query concepts, such as thesauri [Pizzato and de Lima, 2003], or even sources of information available on the Web [Xu et al., 2009]. Particularly with the growth of the Web, and mainly the Web 2.0, characterized by the massive contribution of content by the end users in a direct and collaborative way, there is a constant flux of new information resources with great potential as source of terms for query expansion.

One example is Wikipedia[2], a free-content collaborative encyclopedia to which anyone can contribute. Some studies have shown that many Wikipedia articles have high quality [Giles, 2005], and there are also efforts by its own community and guidelines to help further improving the quality of articles [Wikipedia, 2012]. These efforts, along with the volume of information present in Wikipedia, makes it a potentially valuable resource for Information Retrieval tasks. In fact, Wikipedia has been exploited for automatic query expansion, mainly for queries related to specific entities or narrow and defined topics/concepts [Xu et al., 2009]. In general, defined entities are described in Wikipedia with a single page. Each page corresponds to an article with information and references of relevance for describing the topic or entity of interest. For the task of query expansion, queries with entities/defined topics can be associated with one or more related articles in Wikipedia. These articles can then be used as a 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, several sections, references, infoboxes, etc).

Another possible source of terms for automatically expanding a query is collaborative social annotation (i.e., tags). The act of users annotating Web objects (e.g., documents, photos, videos, etc) has become very popular. Indeed, several recent studies have shown that tags have a great potential to improve services such as automatic object classification [Figueiredo et al., 2009, 2011], searching [Li et al., 2008] and content recommendation [Guy et al., 2010]. Recently, tags have also been exploited as a new resource for extracting relevant terms for query expansion [Lin et al., 2011]. One problem of using tags for these purposes is that they may contain a lot of noise (e.g., misspellings) [Koutrika et al., 2008; Figueiredo et al., 2009]. Moreover, users create and

[2]http://www.wikipedia.org

assign tags with several different purposes other than only describe an entity/concept. For instance, they may use tags for personal organization purposes (e.g., "toread") or for expressing opinions about a webpage (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.

1.1 Objective and Contributions

In this work we propose to combine, expand and improve several of the aforementioned ideas into a new method for expanding queries with references to entities (or narrow concepts/topics). More specifically, we propose a method for automatically filtering, weighting and ranking terms extracted from Wikipedia articles related to the original query. Our method, originally proposed in [Oliveira et al., 2012], is based on state-of-the-art methods for tag recommendation that exploit heuristic metrics to estimate the descriptive capacity of a given term as well as the structure of the Wikipedia page [Belém et al., 2011]. Originally used in 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 Wikipedia article, according to the tag recommendation method, will also be the most relevant ones for the query expansion, given that article and query are semantically related. We believe that the application of state-of-the-art tag recommendation approaches that exploit the aforementioned aspects in order to extract good terms for query expansion is a novel and original contribution of this dissertation.

For evaluation purposes, we compare our proposed method against a state-of-the-art social annotation method [Lin et al., 2011], a state-of-the-art method that also exploits Wikipedia [Xu et al., 2009], and a Language Model PRF method [Lavrenko and Croft, 2001] 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 [Lavrenko and Croft, 2001], [Oliveira et al., 2012]. For all experiments the Indri search engine was used as information retrieval system. Our experimental results indicate 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 14% over the best baseline method. Moreover, these results are close to an ideal oracle that always knows when to expand or not.

In summary, the main contribution of this dissertation is a new unsupervised method of automatic query expansion that exploits the structure of Wikipedia articles to filter, weight and rank candidate terms for expansion. Our approach considers heuristic metrics extracted from Wikipedia articles to estimate the descriptive capacity of candidate expansion terms, ultimately weight and rank those terms in order to use them to improve retrieval performance.

The rest of this dissertation is organized as follows. Chapter 2 reviews related work. Chapter 3 details our proposed method for automatic query expansion. The three collections used in our experimental design are presented in Chapter 4. Chapter 5 reports and analyses the main results obtained. Chapter 6 concludes the work and presents some directions for future work.

Chapter 2

Related Work

Several automatic query expansion methods are available in the literature. Traditionally, pseudo-relevance feedback (PRF) has been the most used of these methods. The basis behind PRF is that documents which are similar to the user's initial query will lead us to more relevant terms which when augmented with the query will lead to an improvement in performance. However, PRF methods have one fundamental problem - query drift. Query drift is caused as a result of adding terms which have no association with the topic of relevance of the query. This happens only when there are only a few or no relevant documents in the top k feedback documents. Due to this sensitivity to the quality of top k documents, PRF only improves the performance of queries which have good or reasonable initial retrieval performance. For instance, [Lavrenko and Croft, 2001] relevance model and [Zhai and Lafferty, 2001] mixture model follow this assumption. The difference between the two methods is that Zhai et al. use this principle for updating the language model instead of only adding terms to the initial query. We here use Lavrenko's relevance model, a *de facto* standard for PRF implemented in the Indri system, as one of our baselines.

More recently, there have been efforts towards improving traditional PRF. In [Metzler and Croft, 2007] the authors proposed a method based on the Markov Random Fields model. The technique, called latent concept expansion, provides a mechanism for modeling term dependencies during expansion. In [Tao and Zhai, 2006], the main idea proposed by the authors is to integrate the original query with feedback documents in a single probabilistic mixture model and regularize the estimation of the language model parameters in the model so that the information in the feedback documents can be gradually added to the original query. Another example of improvement of PRF was proposed in [Cao et al., 2008] in which the author use a classification based approach to selected the terms from the feedback documents. Finally, in [Lee et al., 2008], the

authors proposed a cluster based re-sampling method to select better pseudo-relevant documents based on the relevance model. The main idea is to use document clusters to find dominant documents for the initial retrieval set, and to repeatedly feed the documents to emphasize the core topics of a query.

Other approaches focus on improving query expansion with decision mechanisms based on query characteristics [Cronen-Townsend et al., 2004; He and Ounis, 2007]. That is, these methods disable query expansion when the query is predicted to perform poorly. For example, [He and Ounis, 2007] select the appropriate collection resource for query expansion whereas [Yom-Tov et al., 2005] compare several methods to estimate query difficulty, aiming at applying them to automatic query expansion. As we shall see, our experimental results are very close to those of an oracle that always knows whether to expand.

There have also been some proposals to exploit external sources of information for query expansion. For example, [Cui et al., 2003] proposed a query expansion approach based on mining user logs. The method exploits the correlation between query terms and document terms to select expansion terms for new queries. Another method using logs was proposed by [Fonseca et al., 2005]. In this method, association rules are applied to identify concepts related to a query based on the content of user logs. The motivation for using an external resource is the assumption that query expansion failure can be caused by the lack of relevant documents in the local collection. Therefore, external sources of information may improve the performance of query expansion.

One potential external source for query expansion is Wikipedia. Indeed, [Xu et al., 2009] 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. First, the authors proposed a systematic classification of the queries into entity queries, ambiguous queries and broader queries. Then they evaluated three methods to select terms for the expansion, each modeling the Wikipedia from a different perspective. The authors showed that modest gains can be obtained when expansion is done in a query dependent procedure, being the best results obtained for entity queries. For entity queries the authors firstly ranks all the terms in the entity page, then the top K terms are chosen for expansion. The measure to score each term is defined as: $score(t) = tf * idf$, where tf is the term frequency in the entity page. idf is computed as $log(N/df)$, where N is the number of documents in the Wikipedia collection, and df is the number of documents that contain term t. Our focus is also on entity or narrow queries but not on pseudo-relevance feedback. So we also use this method as one of our baselines.

Other examples of the use of Wikipedia on query expansion can also be cited.

[Brandão et al., 2011] proposed three approaches for extraction of terms related to Wikipedia entities. The first one uses terms available in the textual content of the articles and the other two, called property-based and relationship based, use, respectively, attribute values and references to other entities, both found in infoboxes. Using the structure of the infoboxes this approach takes advantage of the richer semantics implicitly provided by: (i) human editors, which selected the most important terms about an entity to compose the infobox and (ii) the user community, which defines, by means of the infobox template, what information is important to describe entities of a certain class. Also, in [Brandão et al., 2014] proposed a learning to rank approach for entity-oriented query expansion, which considers semantic evidence encoded in the content of Wikipedia articles fields, and automatically labels training examples proportionally to their observed retrievel effectiveness. Experiments on three TREC web test collections attest the effectiveness of the approach, with significantly gains compared to a state-of-the-art entity-oriented query expansion approach. Moreover, by breaking down the analysis by query difficulty, the authors demostrated the robustness of their learning to rank approach when applied for queries with little room for improvement. In addition, they show that the observed improvements hold even when Wikipedia pages are considered in the search results. Lastly, they analyse the performance of multiple sources of semantic evidence separately, showing that statistical and proximity term features are particularly suitable for selecting effective expansion terms.

In [Li et al., 2007] proposed to use the category assignments of Wikipedia articles for the purpose of query expansion. Each query is run against a Wikipedia collection. The number of top-ranked articles is used as weight to each category. Then the articles are re-ranked based on the sum of weights of the categories to which each article belongs. The method shows small improvements over PRF. [Elsas et al., 2008] in turn, investigated link-based query expansion using Wikipedia. They focused on anchor text and proposed a phrase score function. The authors showed that this technique provided significant performance improvements, yielding a 22% and 14% improvement in MAP over the unexpanded query.

Finally, the closest approach to ours, proposed by [Lin et al., 2011], takes advantage of Web 2.0 collaborative social annotations (i.e., tags) for query expansion. It uses a term 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 con-

[1]http://delicious.com

sidered 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. Given its recency and similarities with our approach, in terms of the use of Web 2.0 collaborative external sources, we use the method proposed by [Lin et al., 2011] as our third baseline, further describing it in Section 4.2. Notice that this method relies on supervised learning, while ours is completely unsupervised.

Unlike all these prior efforts, in [Oliveira et al., 2012] we propose an unsupervised tag recommendation based approach for query expansion, which considers heuristic metrics extracted from Wikipedia articles to estimate the descriptive capacity of candidate expansion terms, ultimately weight and rank those terms in order to use them to improve retrieval performance. Experiments ib three TREC web test collections attest the effectiveness of the approach, with gains of up to 14% in terms of MAP compared to a state-of-the-art query expansion approach.

Next, we describe in details our proposed approach for query expansion based on tag recommendation.

Chapter 3

Query Expansion Based on Tag Recommendation

In this Chapter, we describe our method for query expansion, originally proposed in [Oliveira et al., 2012]. It is based in a recently proposed method for recommending relevant tags for a specific Web 2.0 object (i.e., a piece of content such as a video on YouTube) [Belém et al., 2011], [Oliveira et al., 2012]. The tag recommendation method, which is shown to outperform various previous approaches, jointly exploits three dimensions: (i) term co-occurrences with tags previously assigned to the target object, (ii) terms extracted from multiple textual features (e.g., title, description, comments) of the target object, and (iii) several metrics of tag relevance, particularly heuristic metrics that try to capture the capacity of a candidate term to describe the target object.

We here apply this tag recommendation method to the problem of selecting terms for expanding a given query, focusing particularly on dimensions (ii) and (iii). Specifically, we apply the method to filter terms from multiple fields (title, abstract, article sections, references) of a Wikipedia article semantically related to the original query (dimension (ii)). We also use the heuristic metrics proposed in [Belém et al., 2011] (dimension (iii)) to rank the filtered terms with respect to their relevance for the Wikipedia article. The hypothesis is that, given that the Wikipedia article and the 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, leaving to the future the task of exploiting co-occurrence patterns with terms extracted from other fields.

In the following, we first describe the tag recommendation method on which our

approach is based, focusing on the two aforementioned dimensions and then present how it was applied to the query expansion task. (Section 3.2).

3.1 Textual Features and Tag Recommendation

For the task of recommending tags for a target Web object, several sources of information related to the object can be used. In particular, the methods proposed in [Belém et al., 2011] exploit various textual features commonly associated with objects on the Web 2.0. Textual features comprise self-contained blocks of text associated with an object, and usually have a well defined functionality, such as title, description, and comments by users [Figueiredo et al., 2009]. For example, Figure 3.1 shows one example of a web object (a YouTube video) and highlights in red two textual features: title and description of the video.

Generally speaking, the tag recommendation problem can be defined as follows [Belém et al., 2011]: 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 object o, generate a set of candidate tags C_o and recommend the k most relevant tags of this set.

In [Belém et al., 2011], the authors generate candidate terms by exploiting co-occurrence patterns with tags previously assigned to the target object and by extracting terms from other textual features (notably title and description) associated with the target object. They then estimate the relevance of each candidate tag by applying various quality metrics. In particular, the metrics that led to the best recommendation results, outperforming various previous methods, are based on 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). Given their superior performance, we focus on these four heuristics here, describing them next.

The Term Spread of a candidate term c[1] for a target object o, $TS(c, o)$, is defined as the number of textual features of o that contain c, that is:

$$TS(c,o) = \sum_{f_o^i \in F_o} j, \text{ where } j = \begin{cases} 1 & \text{if } c \in f_o^i \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

TS is a metric that exploits the structure of the page, which has only very recently been applied to tag recommendation [Belém et al., 2011], being very different in nature

[1] We disregard stop-words as potential candidates.

Figure 3.1. Web Object from YouTube

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. For instance, if c appears in the title, description, and comments posted by a user of an object o, there is a high chance that c is very related to o's content. The maximum TS value is given by the number of textual features considered (e.g., three if we consider title, description and comments).

The Term Frequency of a candidate c in an object o, $TF(c, o)$, is defined as the number of occurrences of c in all textual features of o:

$$TF(c, o) = \sum_{f_o^i \in F_o} tf(c, f_o^i), \tag{3.2}$$

where $tf(c, f_o^i)$ is the frequency of term c in textual feature f_o^i. In other words, TF considers all textual features of o as a single list of terms, counting all occurrences of c in it. Thus, the main difference between TS and TF is that the former considers the structure of the object in terms of textual features, counting the number of them containing c.

Both TS and TF assume that each textual feature has the same descriptive capacity, which may not be true. For instance, the title of an object may carry terms that more accurately describe its contents than the comments. Then, for truly capturing the importance of each feature, [Belém et al., 2011] proposed two new metrics, namely weighted Term Spread (wTS) and weighted Term Frequency (wTF). These metrics weight a term based on the average descriptive capacities of the textual features in which it appears. This is also an interesting aspect of the metrics that exploit the structure of the object that has demonstrated to be very useful for the tag recommendation task.

The authors estimate the descriptive capacity of a textual feature by the Average Feature Spread (AFS) heuristic [Figueiredo et al., 2009], defined as follows. 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:

$$wTS(c,o) = \sum_{\mathcal{F}_o^i \in \mathcal{F}_o} j, \text{ where } j = \begin{cases} AFS(\mathcal{F}^i) & \text{if } c \in \mathcal{F}_o^i \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

$$wTF(c,o) = \sum_{\mathcal{F}_o^i \in \mathcal{F}_o} tf(c, \mathcal{F}_o^i) \times AFS(\mathcal{F}^i) \tag{3.4}$$

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 of information (see discussion below). Thus, given a query and its related Wikipedia article, our goal is to produce a ranked set of terms that can be used for the purpose of expanding the query. These terms should be closely related to the query, and thus to the Wikipedia article. Towards that goal, we apply the tag recommendation strategy described in the previous section, adapting it to the present purpose. Specifically, the recommendation method is applied to the given Wikipedia article w (i.e., object), taking w's title, summary (first section), content body (all other sections of the article other than the summary) and text of references as the set of textual features F_w of

object (article) w. We generate a set of candidate terms C_w by extracting them from these textual features, and we use one of the descriptive heuristic metrics, i.e., TS, TF, wTS or wTF, to rank these terms according to their relevance to w.

For building each expanded query we use the #weight belief operator of the Indri system [Strohman et al., 2004]. The #weight operator allows more control on the impact of each term on the query for obtaining the final score of the retrieval. In more details, the expanded query is formed with the following structure:

$$\#weight(\delta_{fb} \times Q_{ori}\ (1 - \delta_{fb}) \times Q_{exp}) \tag{3.5}$$

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. In the first one, the Q_{exp} component is composed of only expansion terms, with no weights. In the second one, 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.

For example, let δ_{fb} be 0.5, and suppose that the original query is *euclid*, and the expansion terms are *alexandria, elements, work,* and *mathematics*. In the first, unweighted, expansion approach, the expanded query has the structure: $\#weight(0.5 \times$ *euclid* $(1 - 0.5) \times (alexandria\ elements\ work\ \ mathematics))$

Now suppose that the wTF metric is used as weighting factor, and that the wTF values of *alexandria, elements, work,* and *mathematics* are 5, 3, 2 and 1.5, respectively. In this case, the expanded query has the structure: $\#weight(0.5 \times$ *euclid* $(1 - 0.5) \times$ $(\#weight(5\ alexandria\ 3\ elements\ 2\ \ work\ 1.5\ mathematics)))$

3.3 Entity Queries Identification

Wikipedia is currently the primary source of information for our method, although other Web 2.0 sources with multiple textual features (YouTube, LastFM) can be used in the future for some specific queries (e.g., celebrities, artists). For each query in the test collection, we *automatically* selected those containing a clear and specific entry on Wikipedia using the query classification method proposed in [Xu et al., 2009]. In most cases the selected queries contain entities (e.g., a person or place) or a specific narrow concept (e.g., "dinosaurs"). We call these entity queries.

Wikipedia is organized as one article per topic. So each article summarizes the most relevant information of each topic. In addition to topic pages, Wikipedia contains what is called "redirect" pages which provides alternative ways of referring to a topic:

e.g. the page *Marx* redirects to the article *Karl Marx*. Another resource of Wikipedia is its "disambiguation pages" which list the referents of ambiguous words and phrases that denote two or more topics in Wikipedia.

So for identification of entity queries the corresponding entity page from Wikipedia is the page with the same title field as the query. Queries exactly matching one title of an entity page or a redirect page are classified as Entity Query. Note that queries with both entity page and and disambiguation pages will be defined as Entity Query, because the existing entity page indicates that there is consensus on the dominant sense for the word or phrase.

In the next chapter we describe how we extracted the textual features from the selected entries as well as the coverage of the entities in the evaluated collections.

Chapter 4

Evaluation Methodology

In this chapter we discuss how we evaluated our proposed method, comparing it against state-of-the-art baselines. We describe the collections used in Section 4.1, and introduce the baselines in Section 4.2. Our experimental setup and the metrics used to assess the performance of all methods are discussed in Section 4.3.

4.1 Collections

We here use three well-established TREC collections for evaluating web retrieval quality, namely ClueWeb09 Category B (or simply ClueWeb09B) [Clarke et al., 2009], WT10g [Hawking and Craswell, 2001] and GOV2 [Büttcher et al., 2006]. Table 4.1 summarizes the three collections, presenting the number of queries, the number of entity queries identified with a corresponding Wikipedia page, the respective Wikipedia coverage of the queries (i.e., number of entity queries for which Wikipedia pages were found by the automatic method as described in section 3.3), the number of documents and average query length. For each entity query[1], the corresponding Wikipedia article associated with it was processed, and a list of candidate terms with their respective metrics were generated. We should stress 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 a Wikipedia page in 4.3

A typical Wikipedia article is organized into sections, and each section is a block of text describing an aspect of the information of interest. In general, each article has a

[1]The actual set of entity queries, according to the criteria discussed in Section 3.2, and corresponding Wikipedia pages used in our experiments can be found in
http://dl.dropbox.com/u/84084/clueweb09.txt, http://dl.dropbox.com/u/84084/WT10g.txt, and http://dl.dropbox.com/u/84084/GOV2.txt
for the ClueWeb09B, WT10g and GOV2 datasets, respectively.

title, summary (first section), reference, an infobox, and several other sections related to the entry in question. For our experiments we used the title, summary, the body, and the references as our textual features for term extraction and metric computation. These fields were *automatically* extracted with a special purpose Wikipedia parser designed by us for extracting them. Thus, our method is completely automatic, from the identification of the Wikipedia page (as explained in the previous chapter) up to the extraction of the required fields.

	ClueWeb09B	WT10g	GOV2
Total # of queries	98	100	149
# of entity queries	40	61	120
Coverage	41%	61%	74%
# docs	50,220,423	1,692,096	25,205,179
Avg. query length	1.96	2.31	2.59

Table 4.1. Summary of the Collections

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 [Lavrenko and Croft, 2001]. 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 implementation of the method proposed by [Xu et al., 2009], 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, the method 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 third baseline method is also our implementation of a very recent work described in [Lin et al., 2011]. The authors proposed a method that exploits social annotation systems, such as Delicious, as a source of terms for query expansion. Thus, we here refer to it as Social Annotation-based query expansion, or simply SA. Based on experiments in three TREC collections and in a sample of Delicious, the authors demonstrated that the SA method provides "better terms" for expansion than PRF techniques. To produce "good" terms for expansion, the method first selects and

extracts candidates from Delicious, and then ranks them based on a measure of importance of the term for the query. Although our proposed method could also use the Delicious as source of terms, we focused on Wikipedia because of the better quality of the documents.

To select candidates, the authors consider co-occurrence metrics and three different approaches based on term dependencies: (i) full independence, assuming that query terms are independent, (ii) sequential dependence, assuming dependence between neighboring query terms, and (iii) full dependence, assuming that all query terms are in some way dependent on each other. Since all combinations of query terms should be considered in the full dependence approach, experimental results in fact demonstrated that this approach was unsuitable to select good candidates, since many irrelevant terms would be considered as potential candidates.

Before choosing the candidate terms, the authors first manually filtered out terms which they considered to reflect personal opinions and points of view. The actual candidate selection is based on frequency-oriented co-occurrence scores of a candidate term with query terms in a Delicious page, notably in the tags and document title. The top 100 terms with highest co-occurrence scores in these fields are then selected as candidates. The relevance of each candidate term to the query is then estimated by running new expanded queries consisting of the original query and the candidate, and checking whether the results for the new expanded query produced an increase in Average Precision (AP) larger than a given threshold δ_{thr}.

The actual query expansion is then performed using a supervised method. This method considers a feature-based learning-to-rank algorithm (ListNet) [Cao et al., 2007] and uses *TF*-based, co-occurrence and term popularity measures as features to re-rank the top 100 terms with highest co-occurrence frequencies. Finally, two expansion strategies are considered: (i) to add the top 50 terms in the ranking produced by the learned model, all with the same weight, and (ii) to add the same top 50 terms, but with weights assigned by the learned model. In that work, experiments showed that the use of different weights significantly improved performance only in one of three collections.

As previously mentioned, we implemented our own version of the SA baseline. As we could not get access to the original datasets used in [Lin et al., 2011], we performed our own crawl of Delicious during February 2012. This data sample consists of 10,229,304 tags associated with 560,033 different URLs, with a total of 702,808 distinct tags. Finally, since we also did not have access to the private ListNet algorithm, we used, instead, RankSVM, a public and effective learning-to-rank algorithm [Joachims, 2002]. For term weighting, we used the probability of relevance given by SVM. To

simulate their manual selection, which is not described in details in the original paper, we also removed terms that expressed personal opinions[2] as well as terms that are either too frequent (with more than 300 occurrences) or too rare (with fewer than 10 occurrences), as these are rarely good object (i.e., article) descriptors [Sigurbjornsson and van Zwol, 2008].

4.3 Setup and Evaluation Metrics

We used the Indri 2.6 search engine [Metzler et al., 2004] as our basic retrieval system. Moreover, retrieval effectiveness is measured in terms of Mean Average Precision (MAP) for the top 1,000 documents [Baeza-Yates and Ribeiro-Neto, 2011]. MAP captures not only aspects related to precision and recall, but also the position in which relevant items are returned. Mean Average Precision derives from Average Precision (AP). AP provides a single number instead of a graph. It measures the quality of the system at all recall levels by averaging the precision for a single query:

$$AP = \frac{1}{RDN} \times \sum_{k=1}^{RDN} (Precision\, at\, rank\, of\, k^{th}\, relevant\, document)$$

where RDN is the number of relevant documents in the collection.

Mean Average Precision (MAP) is the mean of Average Precision over all queries. Most frequently, arithmetic mean is used over the query set. We report MAP results for all queries, including non-entity queries (see discussion below), before and after expansion. When selecting terms, we eliminate stop-words, but we 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 methods, we fixed parameters N and k equal to 10 and 50, respectively, and set the $\#weight$ operator parameter δ_{fb} equal to 0.5. These values were chosen as they provided the best results in previous work [Lin et al., 2011]. For the SA baseline, we used the same value for threshold δ_{thr} as the original work, i.e., 0.005. 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

[2] As some of these terms were frequent, a ranking of the terms by frequency helped us to identify most of them. Some additional runs of the SA method helped us to identify other terms based on the recommendations.

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 method as well as the WE baseline, we used two strategies to deal with non-entity queries[4]. In the first one, we kept the original query as it is. In the second strategy, we used the pseudo-relevance feedback. Notice also that in the original proposal of the WE baseline, the authors also treated ambiguous queries (i.e., with more than one possible meaning) and therefore, 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 these 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.

[3]Notice that to some extent, 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

Chapter 5

Experimental Results

In this Chapter, we report MAP results for all considered methods, comparing these results using statistical significant tests (i.e., two-tailed paired Student's t tests) with a 95% confidence level. Specifically, we performed a pairwise comparison of all methods, applying a paired difference test [Jain, 1991] for each pair of methods on each query to verify whether their average results are statistically different with 95% confidence level. To improve readability, we here report only the MAP results, showing in bold the best results, along with statistical ties, for each analyzed scenario.

Recall that we here consider two types of query expansion, with and without weights. We start by discussing the results for unweighted query expansion in Section 5.1. Next, in Section 5.2, we evaluate the performance of the query expansion done with weights assigned to the terms. We then compare our best solution against an oracle that always knows when to expand a query in Section 5.3. In section 5.4, we address the impact of the quality of Wikipedia articles on query expansion effectiveness. Finaly, in section 5.5 we apresent some experiments with the objective of applying machine learning algorithms to combine different methods of query expansion.

5.1 Unweighted Query Expansion

Table 5.1 shows MAP results for each baseline method, namely PRF, SA, WE and for our proposed query expansion strategy. For our strategy, we report the results obtained when each descriptive heuristic metric – TF, TS, wTF and wTS – is used as term ranking criterion. We here focus on the performance of SA, WE and our method without the assignment of weights to terms in the expansion process, deferring to the next section the discussion of results when weights are introduced to the expansion. For PRF, instead, all reported results refer to the method implementation available in

Method	ClueWeb09B	WT10g	GOV2
Orig. Query	0.141	0.195	0.294
PRF	0.141	0.202	0.315
SA	0.142 (0.142)	0.194 (0.199)	0.300 (0.317)
WE	**0.162 (0.165)**	0.183 (0.187)	0.276 (0.282)
TF	**0.170 (0.174)**	0.219 (**0.222**)	0.324 (**0.331**)
TS	**0.168 (0.172)**	0.211 (0.215)	0.318 (0.325)
wTF	**0.169 (0.173)**	**0.221 (0.225)**	0.324 (**0.331**)
wTS	**0.167 (0.171)**	0.212 (0.216)	0.316 (0.323)

Table 5.1. Comparison of Unweighted Query Expansion Methods: MAP Results. Best results for each collection, including statistical ties with 95% confidence level, are shown in bold.

the Indri engine, which does apply weights to terms. As explained, for all expansion methods, we selected the top 50 terms.

Moreover, for our approach as well as for the WE baseline, we report two sets of results: 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 these queries. Similar results are presented for the SA baseline for cases in which there were no Delicious tags co-occurring with the query terms (i.e, no expansion and expansion with PRF). We refer to these two variations of each method as the method *with* and *without PRF*. Best results for each collection, including statistical ties with 95% confidence level, are shown in bold.

As can be observed, 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 [Xu et al., 2009], 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. Moreover, some facts that might explain the better behavior of the WE baseline in the ClueWeb09B collection include: 1) this collection is more recent and perhaps more aligned with the also recent Wikipedia pages we used; and 2) this collection is larger and less focused, better capturing the characteristics of the Web; for the other two collections there may have been some vocabulary mismatch.

On the other hand, our new approach with any of the proposed metrics produces large performance improvements over the original query as well as over all baselines in all three collections. Moreover, there is a slight trend towards some gains from jointly using our method with PRF for non-entity queries over not using PRF: for a given metric and collection, the differences in 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 the ClueWeb collection as all of them lead to MAP results that are statistically tied (with 95% confidence). However, there is a slight tendency for a superior performance, on average, of wTF in WT10g and GOV2. In any case, these results indicate the capacity of all methods of extracting good and relevant terms from the Wikipedia articles, which, in turn, seems to be an excellent source for expansion terms.

In terms of quantitative gains, when we compare our best results with the MAP of the original queries we can observe gains of 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, as performed by the baselines as well as by our proposed method. Table 5.2 shows MAP results for the baselines and for our method with each metric used as both ranking criterion and weight factor as well as for the two approaches to deal with non-entity queries. For comparison purposes, it also shows results for the original query and for the PRF baseline (the same as those shown in Table 5.1). Once again, for all expansion methods, the top 50 terms were selected. Best results for each collection, along with statistical ties (with 95% confidence level) are shown in bold.

We start by noting that, in our experiments, weights did not have impact on

Method	ClueWeb09B	WT10g	GOV2
Orig. Query	0.141	0.195	0.294
PRF	0.141	0.202	0.315
SA	0.143 (0.142)	0.190 (0.195)	0.296 (0.313)
WE	**0.159 (0.161)**	0.182 (0.186)	0.275 (0.282)
TF	**0.168 (0.172)**	**0.225 (0.229)**	0.324 (**0.331**)
TS	**0.166 (0.170)**	0.212 (0.216)	0.319 (0.326)
wTF	**0.167 (0.171)**	**0.225 (0.230)**	0.323 (**0.331**)
wTS	**0.168 (0.172)**	0.213 (0.217)	0.318 (0.325)

Table 5.2. Comparison of Weighted Query Expansion Methods: MAP Results. Best results for each collection, including statistical ties with 95% confidence level, are shown in bold.

the performance of the SA baseline in any collection. These results are in contrast to those reported in the original work [Lin et al., 2011], in which the authors reported a 13% improvement over the version without weights in one of the three collections. We conjecture that this different behavior may be due to the use of a different learner: we here use rankSVM, instead of the private ListNet, which unfortunately is not publicly available. However, we note that our results are not completely inconsistent with the original work, since 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 [Lin et al., 2011]), this would not be enough to outperform our results. The performance of the WE baseline is also not much affected by the weights.

In general, we find that, like with the unweighted version, our approach with any of the four metrics used as both ranking criterion and weight factor produces significant MAP gains over all baselines in all three collections. Most of the gains are very similar to those presented in the previous section. In fact, in ClueWeb09B and GOV2, the weights do not make much difference, as the results are statistically tied with the corresponding unweighed ones. There is a slight tendency of improvement with weights in WT10g, although these gains are not statistically significant with 95% confidence level. The same is observed with the use of PRF for non-entity queries: the results tend to be superior to not using it (particularly in GOV2). We can also see that *wTF* is consistently among the best metrics in all collections, in both the unweighted and weighted scenarios. Recall that *wTF* is one of the most complete metrics that exploit the descriptive capability of the terms as well as the structure of the page into multiple textual fields.

In sum, our proposed method, mainly when using the metric *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.

Table 5.3 shows examples of topics and expansion terms suggested using the *wTF* and ranked according to this metric in ClueWeb09B. The query "the music man", for example, refers to a musical composed by Meredith Willson, while the terms "harold" and marian" refer to main characters of the story. Another interesting example is the topic "obama family tree". The *wTF* metric gave more weight to the terms "malia" and "sasha" (the names of the daughters of US President Barack Obama) than for the term "president", indicating that the metric can capture more precisely the context of the user information need.

Topic	Candidate Terms
the music man	harold marian musical award best broadway willson hill theatre tony
obama family tree	barack, malia, chicago, sasha, president, united, states, robinson
volvo	trucks group company cars global car construction ford heavy
poker tournaments	players chips series buy limit sit world game betting event cash

Table 5.3. Example of Topics and suggested terms using the *wTF* metric in ClueWeb09B

Similar examples are given for WT10g and GOV2 in Tables 5.4 and 5.5, respectively.

Topic	Candidate Terms
sir edmund hillary	tenzing expedition summit everest hunt reached ascent mount
information on j robert oppenheimer	groves project bomb atomic laboratory site scientific military alamos
booker t washington	black education schools rights virginia hampton civil way support slavery leaders
mexican food culture	mexico cuisine chocolate dishes cheese states spanish corn

Table 5.4. Example of Topics and suggested terms using the *wTF* metric for WT10g

Topic	Candidate Terms
ivory-billed woodpecker	world american evidence extinct despite nest grail feathers conservation arkansas
hidden markov modeling hmm	sequence state known variables models possible information continuous
scottish highland games	gaelic scotland dancing events related known athletics entertainment caber toss
pearl farming	oysters water method maturation spat mature seed retrieved cultivation crassostrea

Table 5.5. Example of Topics and suggested terms using the *wTF* metric for GOV2

Method	ClueWeb09B	WT10g	GOV2
Oracle	**0.187**	0.252	**0.364**
WTF with PRF	0.171	0.230	0.331

Table 5.6. MAP Comparison to an *Oracle* that always knows when and when not to expand

5.3 Comparison to an Oracle

As mentioned before, query expansion may be harmful to some queries, mainly for difficult ones (e.g., ambiguous queries). Indeed, we have seen in our experiments some examples of performance losses after query expansion in all collections. Thus, it is interesting to know the potential gain in performance of an ideal method that always knows whether to expand or not. We refer to this method as the *Oracle*.

In more details, we built our Oracle by choosing, for every single query in the collections, the best Average Precision value among three results: expansion with our best method, expansion with PRF, and no expansion at all[1]. Our goal, in this section, is to compare our best results with those produced by this *Oracle* in all three collections.

As shown in Table 5.6, our best results, produced using the WTF metric with PRF, are very close to those obtained with the *Oracle*. Indeed, our method is only around 9% worse than the Oracle in the three collections. These results indicate that even if some queries, after the expansion, have performance losses, very big losses are not often or they are compensated, *on average*, by large gains for other queries. Thus, *on average*, the expansion does bring benefits, particularly if the *wTF* metric is used both as ranking criterion and weight factor for entity queries along with PRF for non-entity queries.

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

5.4 Impact of Quality of the Wikipedia Articles on Expansion

Different Wikipedia articles may have different quality levels in terms of coverage of the topic, length, structure and organization, number of references, etc. Thus, we here briefly investigate whether the quality of the Wikipedia articles used in our evaluation may have affected, to some extent, the effectiveness of the expansion process.

Our analysis is based on the work of [Dalip et al., 2011], which proposes an automatic method for estimating the quality of Wikipedia articles[2]. The authors used a set of 68 features extracted from each article, including features related to the text and organization of the article, network features (i.e., features extracted from the connectivity of the Wikipedia graph), and revision features, along with a Support Vector Regression (SVR) model to estimate article quality. The authors used five quality levels (1-5), which were defined according to Wikipedia standards, going from Featured articles (the best ones) to Stubs (the worst ones). The SVR method, along with the features, tries to estimate such quality level: the closer to 5, the higher the quality of the article.

We applied the method proposed in [Dalip et al., 2011] to the Wikipedia articles selected for the ClueWeb09B queries, considering most of the originally proposed features, except the network features, since we did not have the Wikipedia graph. The five best and worst ranked articles according to that method are shown in Table 5.7 along with the SVR prediction, and the respective Average Precision (AP) of the corresponding query, before and after expansion.

Although a clear pattern may not arise from an analysis of Table 5.7, we find that, in general, there are some very impressive MAP improvements even when the "worst" articles, according to SVR prediction, are used. For instance, the MAP for the query "mitchell college" goes from 0.0168 to 0.2598, after expansion using one of the five worst ranked articles, an increase by a factor of 14. Similarly, we see an improvement by a factor of 28 for the expansion of query "the music man" using one of the five best ranked articles. However, in absolute terms, the AP after expansion of most queries that were expanded using a low quality article is low. But this may have happened because these queries are already very difficult. See, for instance, the very low values of "AP before Expansion" of the last four queries in the table. We intend to investigate this issue further in the near future.

[2]We thank the authors for the availability of their code.

Query	Estimated Quality	AP Before Expansion	AP After Expansion
inuyasha	4.52877	0.2681	0.3534
dinosaurs	4.47195	0.0866	0.2318
toilet	4.09556	0.1388	0.1544
family of barack obama	4.04579	0.3681	0.6172
the music man	3.95726	0.0233	0.6848
french lick resort and casino	2.29291	0.3697	0.4284
mitchell college	1.89317	0.0168	0.2598
orange county convention center	1.74622	0.0578	0.1002
the pampered chef	1.21522	0.0433	0.0897
the current	0.96129	0.0000	0.0037

Table 5.7. Estimation of the Quality of the Wikipedia Article Used for Query Expansion

5.5 Combining Multiple Methods of Query Expansion

In this section we describe our experiments involving machine learning applied to ranking of query terms. Although the results do not show improvement over our metrics, we consider this experiment our first exploration of combining different query expansion methods.

5.5.1 Motivation and Objective

We observed that for several queries some terms were suggested by all methods (our method and baselines). For example, the term "discipleship" was suggested by all methods (TF, TsS, wTF, wTS, SA, WE) for the query "rick warren" (an American evangelical Christian pastor and author). This observation motivated us to try to combine the methods.

As we saw in the previous chapters, each query expansion methods, in our case the proposed metrics and baselines (Wikipedia Entities and Social Annotation), provides pairs (term, weight) where the weight is the estimated relevance of the given term for a given query according to a method.

Our hypothesis is that if a term is suggested by more than one method with high weights, strongest is the evidence that this term is good for expansion. However the weights provided by each method are not comparable, making the task of combining them very difficult. One way to overcome this issue is to use machine learning techniques to learn a function that rank all terms given the different signals of relevance (weights of each method).

5.5.2 Methodology

Learning to rank is a type of supervised machine learning problem in which the goal is to automatically construct a ranking model from training data. The data consists of lists of items with some partial order specified between items in each list. This order is typically induced by giving a numerical or ordinal score or a binary judgment (e.g. "relevant" or "not relevant") for each item.

Traditionally, learning to rank is applied to ranking documents in search engines. The training data consists of queries and documents matching them together with the relevance degree of each match. These query-document pairs are usually represented by numerical vectors, which are called feature vectors. Components of such vectors are called features, factors or ranking signals. Some examples of features are: Term Frequency (TF), Term Frequency - Inverse Document Frequency (TF-IDF), BM25, and language modeling scores of document's zones (title, body, anchors text, URL) for a given query and document's PageRank, HITS ranks and their variants [Baeza-Yates and Ribeiro-Neto, 2011].

Our problem of combining different query expansion methods can also be modeled as a learning to rank problem where for each query we have a list of several terms (instead of documents) and each term has features (the feature vector) describing the estimated relevance (ranking signals) of the term for the query. In our case the feature vector is composed of the weights of each method. And the partial order between terms is induced by a binary judgment ("relevant" or "not relevant") for each item (we detail the judgment process in Section 5.5.2.2). The objective is to rank the terms given a learned function from this training data.

Formally, each query and each of its suggested terms are paired together, and each query-term pair is represented by a feature vector. Thus the training data can be formally represented as: (x_j^q, l_j^q), where q goes from 1 to n, the number of queries, j goes from 1 to m_q, the number of terms for query q, $x_j^q \in \mathbb{R}$ is the d-dimensional feature vector for the pair of query q and the j-th term for this query while l_j^q is the relevance label for x_j^q.

5.5.2.1 Datasets Statistics and Feature Vector Construction

In our context the feature vector has 6 dimensions at most: TF, Ts, wTF, wTS, SA, WE. To identify the features for a given term we performed an exact match comparison: for each term suggested by one method we compared it to all other terms of the other methods. For several terms some features were missing given that they were suggested by different methods using different sources. Table 5.8 shows some statistics of the

terms of each collection while Table 5.9 shows the distribution of the number of features for each collection. Comparing Table 4.1 with Table 5.8 we can see that our coverage for the ClueWeb09B went from 41% to 92%. Also we can see that our metrics (exactly 4 features, TF, TsS, wTF, wTS) dominates the collection and very few instances have all the 6 features.

	ClueWeb09B	WT10g	GOV2
Total # of queries	90	78	132
Coverage	92%	70%	88%
Total # of terms	48374	91927	159325
Minimum number of terms per query	27	6	5
Maximum number of terms per query	4695	3199	4396
Median number of terms per query	100	1044.5	990.5

Table 5.8. Summary of the Distribution of Terms for each Collection

Number of Features	ClueWeb09B	WT10g	GOV2
1	7180	3734	2378
2	–	4	4
4	39433	86015	156208
5	1743	2162	733
6	18	12	2

Table 5.9. Summary of the Features Distribution for each Collection

In Figures 5.1, 5.2 and 5.3 we detail the distribution of number of terms per query for each collection. In particular, the distribution in the ClueWeb collection can be justified by the quality of the terms produced by the SA method: for some queries we were not capable of finding enough terms for expansion.

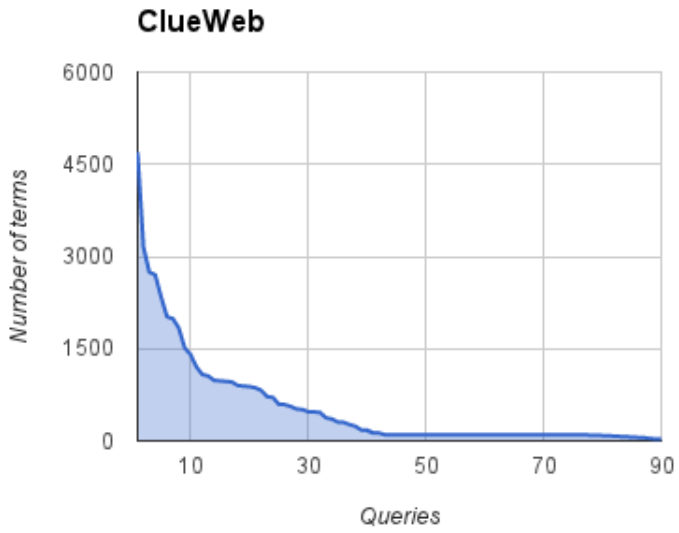

Figure 5.1. Distribution of Number of Terms per Query for the ClueWeb Dataset

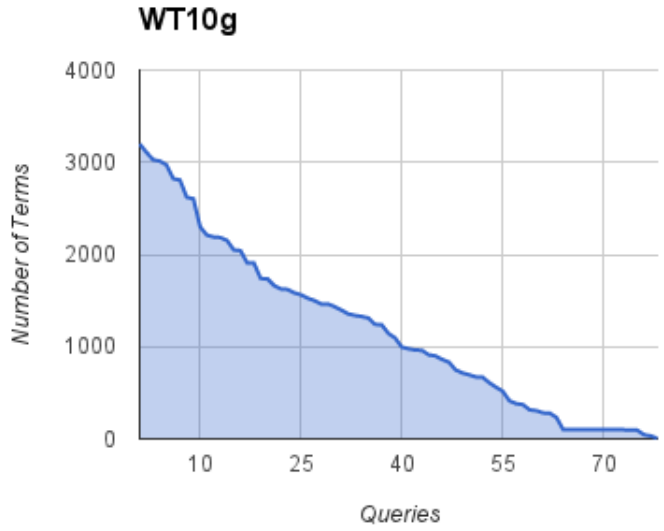

Figure 5.2. Distribution of Number of Terms per Query for the WT10g Dataset

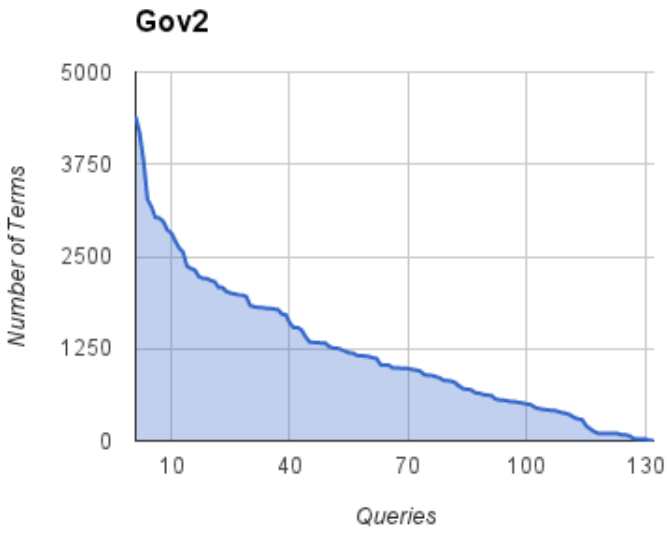

Figure 5.3. Distribution of Number of Terms per Query for the GOV2 Dataset

5.5.2.2 Term Quality Evaluation

For labeling (relevance judgment) the training set we tested each of the candidate expansion terms to check its impact on the retrieval effectiveness. In order to make the test simpler, we made the following simplification: each expansion term is assumed to act on the query independently from other expansion terms. Based on these simplification, we measure the performance change due to the expansion term e by the ratio:

$$chg(e) = \frac{MAP(Q \cup e) - MAP(Q)}{MAP(Q)} \tag{5.1}$$

where $MAP(Q)$ and $MAP(Q \cup e)$ are respectively the MAP of the original query and expanded query (expanded with e).

Now suppose that query q_i has k expansion terms, the relevance label of expansion term e_j $(1 \leq k)$ is defined as follows:

$$label(e) = \begin{cases} 0, & \text{if } chg(e) < 0 \\ 1, & \text{if } chg(e) \geq 0 \end{cases} \tag{5.2}$$

where $label(e_j) = 1$ reflects term e_j is relevant to query q_i, and $label(e_j) = 0$ reflects term e_j is irrelevant.

Table 5.10 summarizes the distribution of relevant terms for each collection while Figures 5.4, 5.5 and 5.6 detail these distributions. As we can see the GOV2 collection

has a long and thin tail which represents that very few queries have a considerable number of relevant terms. This fact may explain the poor performance of expansion in this collection.

	ClueWeb09B	WT10g	GOV2
Total # of queries	90	78	132
Total # of terms	48374	91927	159325
Minimum number of relevant terms per query	0	0	0
Maximum number of relevant terms per query	510	461	1245
Median number of relevant terms per query	6	17	4

Table 5.10. Summary of the Distribution of Relevant Terms for each Collection

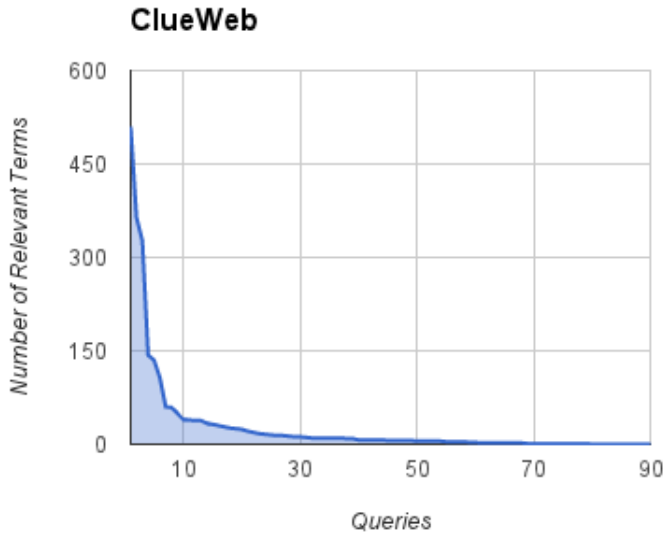

Figure 5.4. Distribution of Number of Relevant Terms per Query for the ClueWeb Dataset

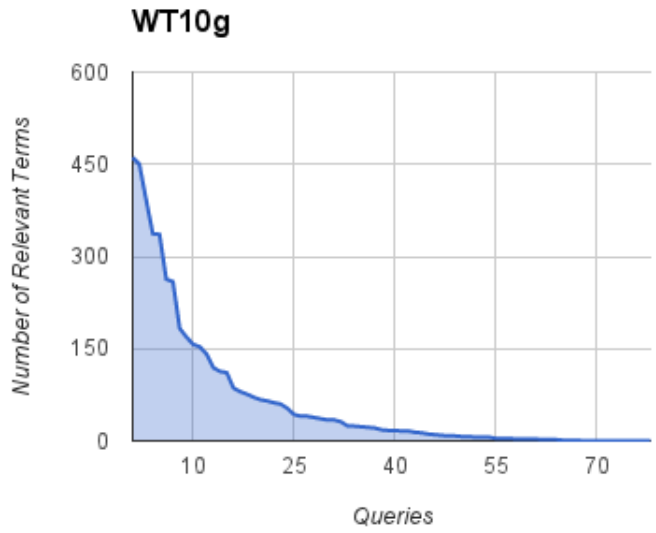

Figure 5.5. Distribution of Number of Relevant Terms per Query for the WT10g Dataset

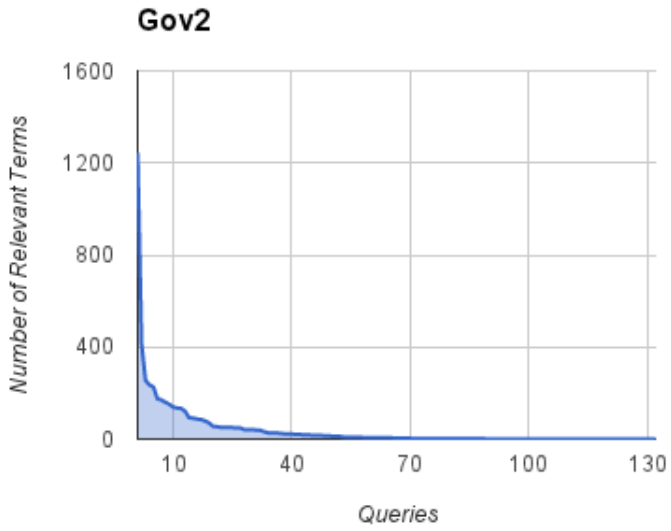

Figure 5.6. Distribution of Number of Relevant Terms per Query for the GOV2 Dataset

5.5.2.3 Learning to Rank Terms

We used k-folded cross-validation to validate the learning approach. In summary, the original dataset is randomly partitioned into k equal size subsamples. Of the k

subsamples, one subsample is retained as a test to the model and the remaining $k-1$ subsamples are used as training data. This process is repeated k times (the folds), with each of the k subsamples used exactly once as testing set. We our case we defined $k = 5$, so each of the test sets represented a subset of all the queries (for each collection).

Two learning to rank algorithms were used: RankSVM and Random Forest. RankSVM [Joachims, 2002] is a pair-wise ranking method adapted from the traditional Support Vector Machines. Pair-wise methods try to order correctly pair of documents by minimizing:

$$\sum_{q} \sum_{i,j,l_i^q > l_j^q}^{m_q} l(f(x_i^q) - f(x_j^q)) \tag{5.3}$$

where l represents a loss function. In the case of RankSVM $l(t) = max(0, 1-t)$.

Random Forests [Breiman, 2001] is an ensemble of tree predictors that outputs the class that is the mode of the classes predicted by the individual trees. Each tree is built according to the following algorithm:

1. Let the number of training cases be N, and the number of variables in the classifier be M.

2. We are given the number m of input variables to be used to determine the decision at a node of the tree; m should be much smaller than M.

3. Choose a training set for this tree by choosing n times with replacement from all N available training cases (i.e., take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.

4. For each node of the tree, randomly choose m variables on which to base the decision at that node. Calculate the best split based on these m variables in the training set.

5. Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

For prediction, a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the mode vote of all trees is reported as the random forest prediction.

The tuning of the RankSVM and Random Forests algorithms was performed using cross-validation in the training set. The Random Forests implementation used was provided by the RankLib `http://people.cs.umass.edu/~vdang/ranklib.html`.

In summary each algorithm outputs a ranked list of terms with associated scores for each query. We then ranked the best terms (the higher the score, the best the ranking) and used them as expansion terms for each query. As in previous experiments, for the queries with no suggested terms, we used the original query for evaluation.

5.5.3 Learning to Rank Results

For evaluating the L2R approach we focused on the unweighted query expansion. Given the output of each algorithm we compared it with the previous methods. Also, we only show the result of the wTF metric as it represented the best overall metric for expansion. For unknown reasons we could not use the Random Forests implementation from the RankLib toolkit in the GOV2 dataset. We validated the dataset with other L2R algorithms from the toolkit but the Random Forests specifically always produced a memory allocation error.

Table 5.11 shows the results for the learning to rank based expansion. As we can see only for the ClueWeb collection the learning based approach produced good results. However these results are still statistically tied with the unsupervised wTF metric. There are some hypotheses for these results. First as we have seen, the distribution of the number of relevant terms is very imbalanced. Very few queries have a sufficient number of relevant terms for learning. Second, the feature distributions are also very imbalanced: very few queries have all the features. A significant number of queries only have one feature produced by the SA method. As we have seen, this method tends to produce very noisy recommendations. However these experiments were very limited on depth and we cannot conclude that it's not possible to apply machine learning to the problem. Even using non-linear machine learning algorithms, we need more features to produce better models (models that can better discimiate relevant from irrelevant terms). And as [Brandão et al., 2014] showed, we already have successful methods of applying machine learning to query expansion.

Method	ClueWeb09B	WT10g	GOV2
Orig. Query	0.141	0.195	0.294
PRF	0.141	0.202	0.315
SA	0.142	0.194	0.300
WE	**0.162**	0.183	0.276
wTF	**0.169**	**0.221**	**0.324**
Rank SVM	**0.170**	0.201	0.301
Random Forest	**0.173**	0.20	-

Table 5.11. Comparison of Unweighted Query Expansion Methods: MAP Results. Best results for each collection, including statistical ties with 95% confidence level, are shown in bold.

Chapter 6

Conclusions

In this work we combined a good source of external information, namely Wikipedia, and an unsupervised state-of-the-art tag recommendation method that exploits the structure of Wikipedia articles into multiple textual fields and descriptive metrics, to produce a method to filter and rank terms for query expansion. To validate and assess the quality of the expanded queries, we ran experiments in three distinct collections and compared our results with three state-of-the-art baselines. In our experimental evaluation, the best consistent 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. In fact, results obtained with this strategy are very close to those produced by an *Oracle* that always knows when to expand. We also performed a brief investigation of the relationship between gains in expansion and the quality of the Wikipedia articles used.

We also investigated how to combine the terms recommended with our methods with some of the lists of terms produced by the WE and SA baselines. Some results in the ClueWeb collection showed that this strategy can produce good results. However we could not generalize to all collections. One possible way to improve the results is a better selection of the features and input dataset. In our experiment we used all queries and features, despite of a significant number of instances with only one feature. Another reason for the poor performance may be the quality estimation for each term. A negative performance of a term evaluated in isolation is not a guarantee that this term will perform poorly in combination with other terms.

There are a lot of avenues to be exploited for future work. One of them is the already mentioned deeper study of the impact of the quality of Wikipedia pages on the expansion process. Other interesting future directions include to exploit other aspects of Wikipedia not considered here, such as infoboxes present in several entity

related pages, links and the taxonomic structure behind entity related pages, redirection and disambiguation information on pages, and to tackle the problem of automatically finding the best Wikipedia pages for a given query. Finally, we also intend to investigate issues related to the size of the expansion list, the performance of the queries, as well as other ways of filtering out bad recommended terms.

Bibliography

Baeza-Yates, R. and Ribeiro-Neto, B. (2011). *Modern Information Retrieval (Second edition)*. Pearson.

Belém, F., Martins, E., Pontes, T., Almeida, J., and Gonçalves, M. (2011). Associative tag recommendation exploiting multiple textual features. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1033--1042.

Brandão, W. C., da Silva, A. S., de Moura, E. S., and Ziviani, N. (2011). Exploiting entity semantics for query expansion.

Brandão, W. C., Santos, R. L. T., Ziviani, N., de Moura, E. S., and da Silva, A. S. (2014). Learning to expand queries using entities. *Journal of the Association for Information Science and Technology*, 65(9):1870--1883.

Breiman, L. (2001). Random forests. *Mach. Learn.*, 45(1):5--32.

Büttcher, S., Clarke, C. L., and Soboroff, I. (2006). The TREC 2006 Terabyte Track. In *Proceedings of 15th Text Retrieval Conference*.

Cao, G., Nie, J., Gao, J., and Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 243--250.

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., and Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129--136.

Clarke, C. L., Craswell, N., and Soboroff, I. (2009). Overview of the TREC 2009 Web Track. In *Proceedings of 18th Text Retrieval Conference*.

Cronen-Townsend, S., Zhou, Y., and Croft, W. (2004). A framework for selective query expansion. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, pages 236--237.

Cui, H., Wen, J., Nie, J., and Ma, W. (2003). Query expansion by mining user logs. *IEEE Transactions on Knowledge and Data Engineering*, 15:829--839.

Dalip, D., Gonçalves, M., Cristo, M., and Calado, P. (2011). Automatic assessment of document quality in web collaborative digital libraries. *ACM Journal of Data and Information Quality*, 2(3):14.

Elsas, J., Arguello, J., Callan, J., and Carbonell, J. (2008). Retrieval and feedback models for blog feed search. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 347--354.

Figueiredo, F., Belem, F., Pinto, H., Almeida, J., Gonçalves, M., Fernandes, D., Moura, E., and Cristo, M. (2009). Evidence of Quality Of Textual Features On The Web 2.0. In *Proceedings of the 18th ACM International Conference on Information and Knowledge Management*, pages 909--918.

Figueiredo, F., Pinto, H., Belém, F., Almeida, J., Gonçalves, M., Fernandes, D., and Moura, E. (2011). Assessing the quality of textual features in social media. *Information Processing and Management*, page in press.

Fonseca, B., Golgher, P., Pôssas, B., Ribeiro-Neto, B., and Ziviani, N. (2005). Concept-based interactive query expansion. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 696--703.

Giles, J. (2005). Special Report: Internet Encyclopedias Go Head to Head. *Nature*, 438(15).

Guy, I., Zwerdling, N., Ronen, I., Carmel, D., and Uziel, E. (2010). Social media recommendation based on people and tags. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 194--201.

Hawking, D. and Craswell, N. (2001). Overview of the TREC-2001 Web Track. In *Proceedings of 10th Text Retrieval Conference*.

He, B. and Ounis, I. (2007). Combining fields for query expansion and adaptive query expansion. *Information Processing Management*, 43:1294--1307.

Jain, R. (1991). *The Art of Computer Systems Performance Analysis - Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley-Interscience, New York.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133--142.

Koutrika, G., Effendi, F., Gyongyi, Z., Heymann, P., and Garcia-Molina, H. (2008). Combating Spam in Tagging Systems. *ACM Transactions on the Web*, 2(4):22:1--22:34.

Lavrenko, V. and Croft, W. (2001). Relevance based language models. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 120--127.

Lee, K., Croft, W., and Allan, J. (2008). A cluster-based resampling method for pseudo-relevance feedback. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 235--242.

Li, X., Guo, L., and Zhao, Y. E. (2008). Tag-based Social Interest Discovery. In *Proceedings of the 17th International Conference on the World Wide Web*, pages 675--684.

Li, Y., Luk, W. P., Ho, K., and Chung, F. (2007). Improving weak ad-hoc queries using wikipedia asexternal corpus. In *Proceedings of the 30th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 797--798.

Lin, Y., Lin, H., Jin, S., and Ye, Z. (2011). Social annotation in query expansion: a machine learning approach. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 405--414.

Metzler, D. and Croft, W. (2007). Latent concept expansion using markov random fields. In *Proceedings of the 30th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 311--318.

Metzler, D., Strohman, T., Turtle, H., and Croft, W. (2004). Indri at TREC 2004: Terabyte track. In *Proceedings of the 13th Text REtrieval Conference*, volume Special Publication 500-261.

Oliveira, V., Gomes, G., Belém, F., Brandão, W., Almeida, J., Ziviani, N., and Gonçalves, M. (2012). Automatic query expansion based on tag recommendation. In

Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12, pages 1985--1989, New York, NY, USA. ACM.

Pizzato, L. and de Lima, V. (2003). Evaluation of a thesaurus-based query expansion technique. In *Proceedings of the 6th International Conference on Computational Processing of the Portuguese Language*, pages 251--258.

Rijsbergen, C. J. V. (1979). *Information Retrieval*. Butterworth-Heinemann.

Santos, R., Macdonald, C., and Ounis, I. (2011). Intent-aware search result diversification. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 595--604.

Sigurbjornsson, B. and van Zwol, R. (2008). Flickr Tag Recommendation Based On Collective Knowledge. In *Proceedings of the 17th International Conference on the World Wide Web*, pages 327--336.

Strohman, T., Metzler, D., Turtle, H., and Croft, W. B. (2004). Indri: A language model-based search engine for complex queries. *Proceedings of the International Conference on Intelligence Analysis*.

Tao, T. and Zhai, C. (2006). Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 162--169.

Wikipedia (2012). Version 1.0 Editorial Team/Release Version Criteria.

Xu, Y., Jones, G., and Wang, B. (2009). Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 59--66.

Yom-Tov, E., Fine, S., Carmel, D., and Darlow, A. (2005). Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *Proceedings of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 512--519.

Zhai, C. and Lafferty, J. (2001). Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th ACM International Conference on Information and Knowledge Management*, pages 403--410.