Contents

Chapter 1

User Generated Content Search

Roi Blanco

Yahoo! Research Barcelona

Manuel Eduardo Ares Brea

University of A Coruna

Christina Lioma

University of Copenhagen

1.1 Introduction

Due to developing technologies that are now readily available, user generated content (UGC) is growing rapidly and becoming one of the most prevalent and dynamic sources of information on the Web. Increasingly more data appears online representing human judgement and interpretation about almost every aspect of the world: discussions, news, comments and other forms of 'socialising' on the Web. The increasing availability of such UGC from heterogeneous sources resembles a *terra incognita* of data and drives the need for advanced information retrieval (IR) technology that enables humans to search and retrieve it, navigate through it, and make sense of it. As such, UGC crosses paths with information retrieval (IR): it creates new IR scenarios, needs and expectations.

This article presents (i) an overview of the main challenges and the respective state-of-the-art (section 1.2), and (ii) a novel and effective approach for using UGC in IR (section 1.3).

1.2 Overview of state-of-the-art

Three representative types of UGC are: *blogs* (short texts), microblogs (short sentences or phrases), and *social tags* (keywords). Each of these poses different challenges to IR, and requires different solutions.

1.2.1 Blogs

Definition 1. A *blog* (Web log) is a discussion or informational Website consisting of discrete entries (*posts*) typically displayed in reverse chronological order (the most recent post appears first). The collection of all blogs on the Web is referred to as *blogosphere*. □

The emergence of blogs in the late 1990's coincided with the advent of Web publishing tools that facilitated the posting of content by non-technical users. Previously, knowledge of such technologies as HTML and FTP had been required to publish content on the Web. As a result of this open publishing paradigm, today the blogosphere is produced by millions of independent bloggers.

Retrieving information from the blogosphere is referred to as *blog distillation*, *blog search*, or *blog feed search* [40]. To facilitate research in this area, in 2006 the Text REtrieval Conference (TREC [64]) started a new track for blog retrieval [50]. This blog track has provided the infrastructure necessary for large-scale evaluation of blog retrieval methodologies: large test collections of blogs with corresponding information needs and relevance assessments.

The blog distillation task is defined as identifying blogs that show a central, recurring interest in a given topic. The task has two main characteristics: (i) the retrieval units are blogs rather than single posts; (ii) in order to be considered as relevant, a blog should not just mention the topic of the user query sporadically, but rather it must contain a significant number of posts concerning this topic. Additional difficulties are posed by these factors: (a) The topics of each post may change over time, hindering the estimation of topical relevance to the query. (b) Posts are time-stamped, so ideally a blog with more recent relevant posts should be ranked higher. (c) Blog posts can have viewer generated comments that can change the relevance of the blog to the query if these are considered as part of the content of the blog.

Different methods have been applied to address these challenges in blog retrieval. We give a brief survey of the state-of-the-art below, focusing on: blog indexing (Section 1.2.1.1), ranking blog posts (Section 1.2.1.2), blog-specific features (Section 1.2.1.3), and document representation (Section 1.2.1.4). More extensive information on blog retrieval can be found in the 2012 survey of Santos et al. [56].

1.2.1.1 Blog indexing

The information needs of users searching the blogosphere fall into two general categories: the need to find individual blog posts regarding a topic, or the need to identify blogs that frequently publish posts on a given topic [25]. These categories mirror the short term versus long term interest distinction observed by Mishne and de Rijke [47] in their study of blog search behaviour. Analogously to this distinction, different blog retrieval approaches may use as indexing unit either (a) an entire blog, or (b) a blog post. The former views a blog as a single document, disregarding the fact that a blog is constructed from multiple posts. The latter takes all or certain samples of posts from blogs and combines their information to produce a single blog score.

When treating each blog as one long document created by the concatenation of all its posts, standard ad-hoc search methods can be used to find relevant blogs to a specific topic. For instance, Efron et al. (2007) take this approach and, given a query, they derive a score for each blog in the corpus using the negative KL-divergence between the query language model and the language model of a blog as a whole [17]. Elsas et al. [19] report an interesting comparison of the two approaches: they experiment with a *large document model* in which entire blogs are the indexing units, and a *small document model* in which evidence of relevance of a blog is harvested from individual blog posts. They also experiment with combining the two models, obtaining the best performance [5].

Currently both approaches are in use; however, Weerkamp (2011) reports that the option of concatenating all blog posts is considered practically unrealistic by most researchers [69].

1.2.1.2 Ranking blog posts

Most approaches rank blogs by making a decision on the relevance scores of all or some of the posts associated with each blog. Different approaches have been proposed.

One idea is to consider blog search as a voting process: A blogger with an interest in a topic is likely to blog regularly about the topic; hence, these blog posts will be retrieved in response to a query topic. Each time a blog post is retrieved in response to a query topic, that can be seen as a weighted vote for that blog to have an interest in the query topic. Then these votes can be combined using data fusion to compute the final relevance score of each blog. This idea has been implemented in a family of voting models [23, 38, 39], which aggregate the relevance score of posts for each blog. These voting models are ported from the area of expert search[1] and effectively treat blogs as the equivalent of experts.

An alternative approach is inspired by the idea that a few posts that are highly relevant to a given topic may sufficiently represent the blog relevance [35]. This approach deals with blog distillation as a resource selection problem, and is mainly inspired by resource selection in distributed information retrieval. Distributed information retrieval uses server selection algorithms to avoid the expensive process of searching all servers for each query [24]. Queries are sent to servers that have more relevant documents to the query. Several studies have ported this idea to blog search by modelling each blog as a collection of posts and then selecting or sampling the best posts [5, 19]. This approach of sampling few relevant posts is reported to outperform using all the posts in the blog [20, 30, 31, 69]. An interesting proposal in this area is the work of Seo and Croft (2008) [58], called *Pseudo Cluster Selection*, where they create topic-based clusters of posts in each blog and select blogs which have the most similar clusters to the query. Also inspired by resource selection approaches, Seo and Croft use diversity penalties: blogs with a diverse set of posts receive a penalty.

1.2.1.3 Blog-specific features

In addition to standard term frequency statistics, a number of other blog-specific features, like user comments and recency, have been been explored for blog retrieval [46, 71], either during re-ranking, or in the first retrieval stage [69, 70]. For instance, Gao et al (2011) [22] explore both heuristic features (e.g., Average Permalink, Sentence Length, Comment number, Organization Numbers, Opinion Rule, etc.) and lexicon-extracted features such as Cyber Words and Cyber Emoticons (using the SentiWordNet and Wilson Lexicon). Elsas et al. (2007) have also used an external resource, namely the Wikipedia, to perform query expansion on blogs [18]. Other approaches include using topic maps [34], or random walks to model connections between blogs, posts and terms [29].

An interesting feature of blogs is their time aspect. Keikha et al. [33] propose a method that uses time-dependent representations of queries and blogs to measure the recurring interest of blogs. In a separate study, they also successfully use the time interval between blog posts when investigating the effect of content-based similarity between posts [32]. Seki et al. (2012)

[1] *Expert Search* is a task in the TREC Enterprise Track, where systems are asked to rank candidate experts with respect to their predicted expertise about a query, using documentary evidence of the expertise found in the collection [61].

also try to capture the recurrence patterns of a blog using the notion of time and relevance [57], while Nunes et al. (2008) use temporal span and temporal dispersion as two measures of relevance over time [48].

Topical cohesion is also another feature that has received attention in blog search. The voting model of Hannah et al. (2007) incorporated *cohesiveness*, defined as how different each post is from the blog as a whole on average. He et al. (2009) [25] proposed a coherence score which captures the topical clustering structure of a set of documents as compared to a background collection. Applied to blogs, this coherence score was found to reflect topical consistency successfully.

1.2.1.4 Blog representations

Blog posts can be represented in different ways. Different approaches use syndicated content (i.e., RSS or ATOM feeds) instead of permalinks (HTML content) [18, 19, 46]; Macdonald et al. (2007) [40] examine whether indexing only the XML feed provided by each blog (and which is often incomplete) is sufficient, or whether the full-text of each blog post should be downloaded. Results of which representation works better are mixed. Other ways of representing documents are, for example, a title-only representation, or an (incoming) anchor text representation. Weerkamp (2011) [69] shows that considering multiple content representations can improve blog search.

1.2.2 Microblogs

Definition 2. A *microblog* is a stream of text that is written by an author over time. It comprises many very brief updates that are presented to the microblog's readers in reverse-chronological order. □

Microblog services let users broadcast brief textual messages online. Twitter is a popular microblogging service that enables its users to send and read text-based messages of up to 140 characters, known as *tweets*. Although microblogging is increasingly popular, methods for organising and searching to microblog data are still relatively new.

Effectively searching microblogs poses a number of issues for traditional retrieval approaches, such as ill-formed language [2], limited document term statistics and spam [21]. The best performing microblog retrieval techniques attempt to utilize both semantic and temporal aspects of documents.

To facilitate research in this area, in 2011 the Text REtrieval Conference (TREC [64]) started a new track for microblog retrieval[3]. This track aims to provide the infrastructure necessary for large-scale evaluation of microblog retrieval methodologies: a test collections of microblogs from tweeter with corresponding information needs and relevance assessments.

[2] http://engineering.twitter.com/2012/05/related-queries-and-spelling.html
[3] https://sites.google.com/site/microblogtrack/

Different methods have been applied to address challenges in microblog retrieval. We give a brief survey below, focusing on: query expansion (Section 1.2.2.1), readily available microblog search engines (Section 1.2.2.2), and microblogs as aids to standard search (Section 1.2.2.3). A more extensive review of microblog search can be found in [15]. See Abel et al. (2011) [1] for more on faceted search for microblogs.

1.2.2.1 Microblog expansion

Several studies have investigated pseudo relevance feedback for microblogs, either using standard textual features (for instance, [14, 41]), or enhanced with temporal information (for instance, [72]). The temporal element of microblogs has also been researched independently of pseudo relevance feedback. For instance, Jabeur et al. (2012) [26] propose a Bayesian network retrieval model that interprets tweet relevance as a conditional probability and estimates it using text similarity measures, the microblogger's influence, the time magnitude and the presence of hashtags. In another interesting study, Metzler et al. (2012) [43] focus on microblog search of past events over microblog archives. Rather than retrieving individual microblog messages in response to an event query, they propose retrieving a ranked list of historical event summaries by distilling high quality event representations using a novel temporal query expansion technique. Specifically, their method takes a query as input and returns a ranked list of structured event representations. This is accomplished in two steps, timespan retrieval and summarisation. The timespan retrieval step identifies the timespans when the event happened, while the summarisation step retrieves a small set of microblog messages for each timespan that are meant to act as a summary.

Instead of query expansion, Efron et al. (2012) [16] propose a document expansion model for microblogs that models not only lexical properties, but also temporal properties of documents. Finally, Liang et al (2012) [36] present an approach that models for the query and the document and combine it with temporal re-ranking.

1.2.2.2 Microblog search engines

Even though microblog search is not one of the oldest UGC search tasks, there already exist several out-of-the-box engines, tailored to this type of content. For instance, QuickView [37] is microblog search platform, which includes Natural Language Processing (NLP) functionalities, such as tweet normalisation, named entity recognition, semantic role labeling, sentiment analysis, tweet classification, etc. It also includes several interface options (such as clustering search results) to facilitate the display and interaction of the user with the information retrieved.

Another example is Twinder [62], which implements features such as topic - tweet semantic relatedness, as well as syntactic, semantic, sentiment and contextual properties of the microblogs. The team behind Twinder has also created Twitcident [2], a framework for filtering, searching and analysing microblog information about real-world incidents or crises. This framework is currently used with data from emergency broadcasting services in the Netherlands, however the technology powering it can be ported to other languages.

The actual search engine powering Twitter's own search is Early bird [10]. Busch et al. (2012) describe its indexing structure and operations, pointing to the important challenge of dynamically ingesting content and making it searchable immediately, while also concurrently supporting low-latency, high-throughput query evaluation. Related to this issue of efficiently indexing real-time microblogs is the work of Bahmani et al. (2012) [7], who present a partitioned multi-indexing scheme for efficient microblog indexing.

1.2.2.3 Microblogs as aids to standard search

In addition to searching among microblogs, the content of microblogs can also be used to facilitate standard document search. Shuai et al. (2012) [60] present an approach that uses information from tweets to rerank news search. They propose a Community Tweets Voting Model (CTVM) that effectively reranks Google and Yahoo! news search results on the basis of open, large-scale Twitter community data.

A different approach is proposed by Rowlands et al. (2010) [55]. The authors present a Web search system based on hyperlinks retrieved from microblogs. When a Twitter message contains a URL, they use the Twitter message as a description of the URL's target, i.e. as an additional form of annotation. Their method is shown to be effective in improving overall retrieval performance.

1.2.3 Social Tags

Definition 3. A *social tag* is a non-hierarchical keyword or term assigned to a piece of information (such as an Internet bookmark, or digital image). This kind of metadata helps describe an item and allows it to be found again by browsing or searching. Tags are generally chosen informally and personally by the item's creator or viewer. □

Early forms of collaborative tagging can be traced back to medieval times, when manuscripts were annotated before being passed down to generations in a 'cumulative scholarship' process [27]. That type of annotation was considered complementary to the scholarly value of the manuscripts and hence was thought to augment the value of the manuscripts. Similarly to that, the type of collaborative tagging found on the Web today can also be seen as a kind of hypermedia augmentation [9]. Furthermore, the notion of annotating the Web is not new: in NCSA Mosaic, one of the earliest browsers, Web pages could be privately tagged with tags being stored on the user's machine [9]. Independently to NCSA Mosaic, the idea of asking users to tag text freely was initially developed by Hidderley and Rafferty (1997), who aimed to index particularly subjective forms of information where full-text searching was failing, such as multimedia or fiction objects. Hidderley and Rafferty (1997) developed the idea of aggregating users' indexing terms to create a generalised overall view of the resources, which today has been adapted by working collaborative tagging systems. The plethora of current collaborative tagging systems cover different domains (e.g. CiteULike, Connotea[4]) and media (e.g., YouTube, Last.fm[5]) as well as different applications (e.g. integrated into enterprise search like ConnectBeam, or recommender systems like Amazon[6]).

The field of IR has also shown interest in collaborative tagging from 2006 onwards [47]. Several commercial IR systems now include recommendation functionalities based on user tags, e.g. Amazon uses tags to suggest relevant products to online buyers. In addition, studies investigate analogies between users-products in recommender systems and queries-documents in IR systems [45, 68]. We briefly overview uses of social tags for IR, focussing on (i) social tags as aids to text search (section 1.2.3.1), and (ii) social tags as aids to image search (section 1.2.3.2).

1.2.3.1 Social tags for text search

The motivation behind using social tags for textual IR is to try and induce the extra information (user perspective, opinion, assessment) represented in it, as a potentially valuable source of information about the relevance between a query and a document. For instance, Bao et al. (2007) [8], proposed optimising Web IR with tags in two ways: Firstly, they used tags as an indicator of Web page popularity, and secondly, they computed the similarity between tags and queries. They incorporated both types of information into document ranking, in two variants of PageRank called SocialPageRank (SPR) and SocialSimRank (SSR) respectively. They reported significant improvements in performance, using real-world datasets (50 manual queries and 3000 automatically generated queries on a dataset crawled from Delicious).

A different approach was proposed by Zhou et al. (2008) [73]. They used tags to enhance document ranking in Web IR as follows: topics are modelled in documents and also in tags, and then the information of the tag topics is incorporated into retrieval as a Bayesian Inference Network. Significant improvements in retrieval performance are shown over traditional approaches.

Jin et al. (2009) [28] propose a query expansion technique that uses social tags to expand queries. They fetch and filter social tags from Delicious, and then compute their similarity to the query terms, before using them for query expansion.

Finally, Carman et al. (2008) [11] investigated the use of tag data for evaluating personalised retrieval systems involving thousands of users. Using data from the social bookmarking site Delicious, they effectively rated the

[4]http://www.citeulike.com, http://www.connotea.org
[5]http://www.youtube.com, http://www.lastfm.com
[6]http://www.connectbeam.com, http://www.amazon.com

quality of personalised retrieval results. They also reported that user profiles based on the content of bookmarked URLs are generally superior to those based on tags alone.

1.2.3.2 Social tags for image search

For similar motivations to above, social tags have also been used to enhance the effectiveness of image search. These approaches are however also driven by the general sparsity of textual annotations among most image collections.

van Zwol et al. (2008) [63] use social tags for improving large-scale image retrieval on the Web. They propose a query model that is estimated from the distribution of social tags, so that the dominant sense of the query is enhanced. They find that social tags are particularly useful in the case of ambiguous queries.

Pedro et al. (2012) [52] propose another approach for exploiting the similarity between the query and the social tag metadata of images. They use social tags to infer an aesthetic rating for the images, which can enhance overall retrieval performance. A user study involving 58 participants confirms the effectiveness of social tags as an aesthetic predictor for images.

1.3 Social Tags for Query Expansion

The previous sections overviewed the main challenges and the respective state-of-the-art in UGC retrieval. Next we present a novel approach for using UGC to assist standard search in the form of query expansion (section 1.3). This method differs from the approach of Jin et al. [28] on several points, which are detailed in section 1.3.2. Section 1.3.1 describes our model, and section 1.3.2 presents and discusses the experimental evaluation.

1.3.1 Problem Formulation

We now introduce a model for expanding queries with salient social tags. The aim is to approximate the informative salience of τ. We posit that the more informative a social tag τ is, the more useful it may be as a query expansion term when estimating the relevance of a document d to a query q.

We estimate tag salience using three different robust term weighting schemes: (i) an adaptation of the Inverse Document Frequency (IDF) scheme; (ii) an adaptation of the Term Frequency - Inverse Document Frequency (TF-IDF) scheme; and an adaptation of the probabilistic Bose 1 (Bo1) scheme from the Divergence from Randomness (DFR) framework [4]. Each of these weighting schemes computes a tag weight $\mathrm{w}(\tau)$ which represents the estimated significance of the tag.

Using IDF, $\mathrm{w}(\tau)$ is computed as follows:

$$\mathrm{w}(\tau) = \log_2 \frac{N}{n_\tau} \tag{1.1}$$

where N in the number of documents in the collection and n_τ is the number of documents that contain τ.

We extend Equation 1.1 to compute a TF-IDF variation of tag significance as follows:

$$\mathrm{w}(\tau) = \tau f_x \cdot \log_2 \frac{N}{n_\tau} \tag{1.2}$$

where τf_x is the tag frequency in the top x (pseudo relevant set of) documents.

We use Bo1 to estimate the divergence of the tag occurrence in the pseudo relevant set of documents from a random distribution of documents. Bo1 is based on Bose-Einstein statistics, and has been shown to be similar to Rocchio [54]. Using Bo1 we estimate the weight of a tag $\mathrm{w}(\tau)$ as:

$$\mathrm{w}(\tau) = \tau f_x \cdot \log_2 \frac{1 + P_n}{P_n} + \log_2(1 + P_n) \tag{1.3}$$

where $P_n = \frac{F}{N}$ where F is the frequency of the tag in the collection. Equation 1.3 is the same as Amati's original Bo1 formula with the sole difference that, in the original, he used tf_x (term frequency in the top x retrieved documents), whereas we use τf_x (tag frequency in the top x retrieved documents). The complete derivation of the Bo1 formula, going back to first principles, is presented by Amati [4].

Next we present how we use the tag weights computed with Equations 1.2 and 1.3 for query expansion. First we give the general ranking formula:

$$\mathrm{R}(d, q) = \sum_{t \in q} \mathrm{w}(t, d) \cdot \mathrm{w}(t, q) \tag{1.4}$$

where $\mathrm{R}(d, q)$ is the approximation of the relevance between the document and the query, t is a term in q, $\mathrm{w}(t, d)$ is the weight of term t for a document d, and $\mathrm{w}(t, q)$ is the query term weight. $\mathrm{w}(t, d)$ can be computed by different weighting models in different ways, for instance using BM25 [53]. $\mathrm{w}(t, q)$ can be computed as: $\mathrm{w}(t, q) = \frac{\mathrm{qtf}(t,q)}{qtf_{max}}$, where $\mathrm{qtf}(t, q)$ (qtf_{max}) is the term frequency (maximum term frequency) in the query. Very often, especially in short queries, $\mathrm{qtf}(t, q)$ will be equal to qtf_{max}, and hence $\mathrm{w}(t, q) = 1$.

We integrate the tag weights into the estimation of $w(t, q)$ as follows:

$$\mathrm{w}(t, q) = \alpha \cdot \frac{\mathrm{qtf}(t, q)}{qtf_{max}} + (1 - \alpha) \cdot \frac{\mathrm{w}(\tau)}{w_{max}(\tau)} \tag{1.5}$$

where $\mathrm{w}(\tau)$ is the tag weight computed with Eq. 1.1-1.3, $w_{max}(\tau)$ denotes the maximum among $\mathrm{w}(\tau)$, and α is a smoothing parameter ($0 < \alpha \leq 1$) to control the balance between of the old and the new weights.

Equation 1.5 expands queries only with social tags. In order to expand queries with both social tags and also traditional expansion terms, we use:

$$\mathrm{w}(t,q) = \alpha \cdot \frac{\mathrm{qtf}(t,q)}{qtf_{max}} + (1-\alpha) \cdot \frac{\mathrm{w}(\tau)}{w_{max}(\tau)} + \frac{\mathrm{w}(t)}{w_{max}(t)} \tag{1.6}$$

where $\mathrm{w}(t)$ is the new weight of a term in the expanded query, and $w_{max}(t)$ is the maximum $\mathrm{w}(t)$ of the expanded query terms. If the expanded term was not in the original query, then $\mathrm{qtf}(t,q) = 0$. Note that the term-PRF terms and weights are obtained using the original query q and not the query expanded with tag-PRF.

1.3.1.1 Jin's method

In [28] Jin et al. propose a query expansion approach which uses collaborative tags, which we have used as baseline in this chapter.

Given a query q to expand composed by k query terms $q_1, q_2, ..q_k$ the first step of their method is querying the collaborative on-line tagging system (Delicious in this case) with each of the individual terms, obtaining k results sets (lists of tagged documents r) $L_1, L_2, ..L_k$. After cleaning the obtained tags by manually removing ill-formed tags and splitting ill-formed compound-tags with the help of a dictionary they calculate a score for each remaining tag t in any of the L_k in order to determine which tags will be used in query expansion. This score is calculated as:

$$\mathrm{score}(t,q|C) = \sum_{q_j \in q} \mathrm{idf}(q_j,C)\ \mathrm{idf}(t,C)\ \log(\mathrm{codegree}(t,q_j|L_j)+1) \tag{1.7}$$

where $\mathrm{idf}(q_j,C)$ and $\mathrm{idf}(t,C)$ respectively measure the uncommonness of the query term q_j and the tag t in C, the union of all the L, and are calculated as:

$$\mathrm{idf}(\alpha,C) = \log \frac{|C|}{\mathrm{df}(\alpha,C)+1} \tag{1.8}$$

and being $\mathrm{codegree}(t,q_j|L_j)$ a measure of the relation between t and q_j, calculated as:

$$\mathrm{codegree}(t,q_j|L_j) = \frac{\sum_{r \in L_j}(\log(\mathrm{tf}(t,r)+1)\ \log(\mathrm{tf}(q_j,r)+1))}{|L_j|} \tag{1.9}$$

Once the scores have been calculated the t_{Jin} tags with the highest score are used to expand the original query q creating a new query q'. The weight of each of the terms of this expanded query (composed by the selected tags as well as the original terms of q) is calculated as:

$$w(q'_m, q') = \alpha \cdot w(q'_m, q) + (1 - \alpha) \cdot \frac{\text{score}(q'_m, q|C)}{score_{max}} \tag{1.10}$$

where $score_{max}$ is the maximum score and $\alpha \in [0, 1]$ is a parameter which controls the balance between the original and the expanded weights.

Using this approach, Jin et. al report consistent gains in MAP and P@10 on TREC 2008 feedback data (264 queries) over the non expanded queries, although they do not compare against any other query expansion approaches.

1.3.2 Experimental Evaluation

1.3.2.1 Methodology and Settings

Methodology

1. Given a query q and an expansion collection C_{tag}, we retrieve a ranked list of tagged documents L_{tag} in response to the whole query q. C_{tag} is an online dynamic collection of collaboratively tagged documents, for which we have no prior knowledge of relevant or non-relevant documents to our query. We have no knowledge of the statistics of that collection either. Retrieval takes place online using a freely-available IR system that supports collaborative tagging, which we cannot modify, but only use as a black-box. Furthermore, we disregard any knowledge of the ranking function of that system.

2. We collect the t_{tag} tags which appear in more documents in L_{tag}. The tags collected in this way for a query are denoted T_L.

3. We expand the initial query q with the tags in T_L associated to q, and we weight these tags according to their significance.

4. The expanded query is used to retrieve a new ranked list of documents L' from a retrieval collection C. C is a static collection (e.g. TREC collection) for which we have prior knowledge of relevant and non-relevant documents to a given query in order to perform evaluation. Retrieval takes place offline using a ranking function that we can modify.

Tag selection. We collect social tags from Delicious and YouTube, two popular online systems where users collaboratively tag resources, respectively Web-links and videos.

Note that, even though we use Delicious and YouTube as a black-boxes in order to collect tags, tags are then processed and weighted according to the method used for tag-PRF.

Hence, both methods do not use the tags for expanding a query straight

TABLE 1.1: Characteristics of the four TREC datasets used. These characteristics vary across collections.

Collection	**LA Times**	**WT2G**	**BLOG06**
Queries	401-450	401-450	901-950
Size	475MB	2GB	25GB
Documents	131,896	247,491	3,215,171
Terms	189,545	1,002,586	4,968,020
Task	Ad-hoc news	Ad-hoc Web	Blog IR
Year	1989-1990	1997	2005-2006

out of the black-box, but they compute their significance and use them accordingly. It should be noted also that the process used to gather the tags is different in each case, as each method proposes a different treatment of the original query in order to query the online systems. In our case, and because we collect tags from online black-box systems, the collection statistics are not publicly available to count statistics of tags over Webpages (certain collaborative tagging systems, such as Delicious, provide the most popular urls and tags only). Hence, we measure the collection statistics of the retrieval collection instead of the tagged collection. This is supported by studies showing that there are similarities in the vocabulary used for tagging and for searching content on the Web [12].

Datasets. We use three standard TREC collections of different domain, size, and timeframe, namely Los Angeles Times (LAT) (475MB), WT2G (2GB), and BLOG06 (25GB). LAT represents a sampling of approximately 40% of the articles published by the Los Angeles Times during 1989-1990, hence it is assumed to be fairly homogeneous. WT2G contains text crawled from the Web in 1997. BLOG06 is a blog crawl of 753,681 feeds, and associated permalink and homepage documents, resulting in approximately 3 million documents from late 2005 and early 2006.

WT2G and BLOG06 are representative of everyday language found on the Web, which is itself a heterogeneous source. Overall, the three collections belong to different domains (journalistic, everyday Web, blog). The size of the collections also differs significantly (475MB-25GB).

For each collection, we use its associated set of queries, shown in Table 1.1. We experiment with short queries (title portion) only, because they are more representative of real user queries on the Web [51].

These datasets have been used in three different TREC retrieval scenarios, namely ad-hoc search (LAT), Web search (WT2G), and blog search (BLOG06). By using them, we aim to test the applicability of our technique in these scenarios.

Retrieval settings. For retrieval we use the Terrier IR system [49]. We match documents to queries with an established and widely-used model, Best Match 25 (BM25) [53]. BM25 computes the relevance of a document d to a query q as:

$$\mathrm{R}(d,q) = \sum_{t \in q} \log\left(\frac{N-n+0.5}{n+0.5}\right) \cdot \frac{(k_3+1)\cdot \mathrm{qtf}(t,q)}{k_3+\mathrm{qtf}(t,q)} \cdot \mathrm{tfn}(t,d) \tag{1.11}$$

where k_3 is a parameter, $\mathrm{qtf}(t,q)$ is the query term frequency, N is the number of all documents in the collection, n is the number of documents containing term t, and $\mathrm{tfn}(t,d)$ is the normalised term frequency in a document, given by:

$$\mathrm{tfn}(t,d) = \frac{(k_1+1)\cdot \mathrm{tf}(t,d)}{\mathrm{tf}(t,d)+k_1\cdot(1-b+b\cdot\frac{l}{l_{avg}})} \tag{1.12}$$

where k_1 and b are parameters, $\mathrm{tf}(t,d)$ is the term frequency in the document, and l (l_{avg}) is the document length (average document length in the collection).

For weighting the expanded query terms in the second pass retrieval, we use the following models:

For standard term-PRF, we use the original Bo1 formula [4]:

$$\mathrm{w}(t) = tf_x \cdot \log_2 \frac{1+P_n}{P_n} + \log_2(1+P_n) \tag{1.13}$$

where $\mathrm{w}(t)$ is the term weight to be computed, tf_x is the term frequency in the top x documents used for PRF, and $P_n = \frac{F}{N}$, where F is the term frequency in the collection, and N is the number of documents in the collection.

For tag-PRF, we use Jin et al.'s method and the method proposed in this paper. In the former the weighting method is summarised in section 1.3.1.1, Equations 1.7-1.10, whereas in the later the IDF, TF-IDF and Bo1 extensions proposed in Section 1.3.1, Equations 1.1-1.3 respectively are used.

Evaluation measures. We evaluate retrieval performance using Mean Average Precision (MAP) and Precision at 10 (P@10).

1.3.2.2 Parameter Tuning

Our experiments will vary the following parameters:

- BM25(Equation 1.11) includes k_1 and k_3, which have little effect on retrieval, and b, which normalises the relevance score of a document for a query across document lengths [53].

- Term-PRF includes t_x, which is the number of terms used for PRF, and d_x, which is the number of top-retrieved documents used for PRF.

- Jin et al.'s PRF includes two parameters, t_{Jin}, the number of terms using for PRF, and α, which controls the balance between the original and the expanded weights.

- Tag-PRF includes the same two parameters, t_{tag}, the number of terms using for PRF, and α, which controls the balance between the original and the expanded weights.

Note how the three PRF methods used in the experiments have all two parameters, in order to keep the comparison between them as fair as possible. For instance, we have chosen not to tune also the amount of documents used in the case of the Tag-PRF approaches, using directly the ones returned by each query to Delicious or YouTube instead. For each method the values of its two parameters are estimated heuristically by tuning in order to select the optimal values to use for PRF. This kind of heuristic tuning is standard in the area [4].

We perform experiments with tuned and cross-validated parameters separately.

Tuned. We tune parameters so as to optimise MAP on the basis of the corresponding relevance assessments available for the queries and collections employed as follows.

- For BM25, we vary b between 0-1, in 10 intervals of 0.1. We do not tune k_1 or k_3 because they have little effect on retrieval performance.
- For term-PRF with Bo1, we vary $t_x \in \{1, 2, ..50\}$, and $d_x \in \{1, 2, ..10\}$.
- For Jin et al.'s PRF, we vary $t_{\text{Jin}} \in \{1, 2, ..50\}$ and $\alpha \in \{0.1, 0.2, ..0.9\}$
- For our tag-PRF, we vary $t_{tag} \in \{1, 2, ..50\}$ and $\alpha \in \{0.1, 0.2, ..0.9\}$

In the cases involving PRF, the parameters were tuned using Coordinate Ascent [42], a technique which iteratively solves alternatively one-dimensional searches in each parameter until convergence. For example, let us take the example of optimising term-PRF. Starting from certain initial values for BM25's b and term-PRF's t_x and d_x, Coordinate Ascent proceeds by first testing all possible values of b, obtaining and saving the one which yields the best performance (in our case the best MAP). Afterwards, all the possible values of t_x are tested for that value of b and the initial value of d_x, saving the one providing the best MAP. Then, all the possible values of d_x are tested using the saved values of b and t_x. This value is saved and the process starts again, this time with the saved values.

When using both term-PRF and tag-PRF (the second experiment) only b is tuned; the parameters of the two PRF methods are the ones obtained by tuning each of them on their own for experiment 1. For instance, the "Tuned" value of term-PRF + Jin et al. PRF is obtaining using the t_x and d_x obtained when tuning term-PRF alone and the t_{Jin} and α obtained when tuning Jin et al.'s method alone.

Cross-validation. The cross-validated performance values were obtaining by using the values of the parameters tuned in the corresponding collection. The cross-validated values for collection LAT and BLOG06 are those tuned in

WT2G, whereas the ones used in WT2G are the parameters cross-validated in BLOG06. Hence, the reported cross-validated MAP and P@10 values for term-PRF in BLOG06 use the b, t_x and d_x parameters tuned for term-PRF in WT2G.

1.3.2.3 Findings and Discussion

We now highlight the main outcome of the experiments. First, we discuss how the proposed method measure up with existing approaches, comparing the performance of pseudo relevance feedback using terms, using social tags with that of the baseline method without expansion.

Table 1.2 (MAP) and Table 1.3 (P@10) report on the results of the two flavours of pseudo relevance feedback and the baseline methods alone. The method that uses social tags (*tag-PRF*) is able to improve consistently the No-PRF baseline method, being the differences more remarkable when tags are retrieved from the YouTube service. This might be due to the fact that this service has a higher number of tags, which in turn results in higher quality tags. When comparing the two social-tag expansion methods, Jin et al.'s has a slight advantage in some of runs on the Delicious collection, but underperforms when using the larger collection to retrieve tags (YouTube). In any case, *tag-PRF* provides the best overall results on both collections. The comparison with respect to term-PRF is somewhat mixed, being the numbers of WT2G and BLOG06 comparable. However, on the LATimes collection, term-PRF is consistently better. This is due to the odd nature of this collection and the vocabulary mismatch between the social tags collections, which are newer and, in general, return results (tags) that are not present as terms in LATimes. Furthermore, the differences between the three weighting schemes (idf, tf-idf and Bo1) are negligible in most of the cases.

Finally, we address the question of whether we can combine terms and tags successfully for PRF or not. Tables 1.4 (MAP) and 1.5 (P@10) compare retrieval performance using a combination of tags and terms for PRF. The combination outperforms best single PRF at most times and is consistently better than the baseline, except for the LATimes collection. This indicates that tag-PRF contributes something extra to term-PRF, which benefits retrieval performance. The fact that tags are beneficial for retrieval is particularly apparent in the newer BLOGS06 collection when using the YouTube tags, where adding terms to tags actually decreases the quality of the results (it should be noted however that the values of the parameters are not optimised for the term+tag combination). This indicates that both PRF approaches are able to improve overall retrieval performance (note that all the scores in the Tables outperform the retrieval model baseline). Earlier observations regarding collections and retrieval models remain valid here. BLOG06 benefits more than the other collections and MAP benefits more that P@10. This consistency indicates that the combination of the two PRF methods is successful and does not disrupt retrieval in any way.

TABLE 1.2: No PRF versus with term-PRF versus tag-PRF. Performance shown in MAP with Tuned and Cross-validated settings, using tags from Delicious (Del) and YouTube (Yt).

			LAT				
				Jin et al.		**tag-PRF**	
	no PRF	**term-PRF**		**tag-PRF**	**Idf**	**TfIdf**	**Bo1**
Tuned	0.2566	**0.3033**	**Del**	0.2678	0.2613	0.2572	0.2572
			Yt	0.2598	0.2809	0.2720	0.2718
Crossval	0.2462	**0.2912**	**Del**	0.2525	0.2539	0.2477	0.2476
			Yt	0.2554	0.2748	0.2695	0.2643
			WT2G				
				Jin et al.		**tag-PRF**	
	no PRF	**term-PRF**		**tag-PRF**	**Idf**	**TfIdf**	**Bo1**
Tuned	0.3167	**0.3427**	**Del**	0.3233	0.3239	0.3225	0.3228
			Yt	0.3213	0.3413	0.3414	0.3400
Crossval	0.3167	0.3266	**Del**	0.3210	0.3222	0.3197	0.3180
			Yt	0.3185	0.3348	**0.3384**	0.3366
			BLOG06				
				Jin et al.		**tag-PRF**	
	no PRF	**term-PRF**		**tag-PRF**	**Idf**	**TfIdf**	**Bo1**
Tuned	0.3487	0.3601	**Del**	0.3516	0.3490	0.3490	0.3487
			Yt	0.3588	0.3722	**0.3760**	0.3754
Crossval	0.3487	0.3337	**Del**	0.3500	0.3489	0.3450	0.3460
			Yt	0.3577	0.3625	**0.3723**	0.3662

TABLE 1.3: No PRF versus with term-PRF versus tag-PRF. Performance shown in P@10 with Tuned and Cross-validated settings (for MAP), using tags from Delicious (Del) and YouTube (Yt).

			LAT				
				Jin et al.		**tag-PRF**	
	no PRF	**term-PRF**		**tag-PRF**	**Idf**	**TfIdf**	**Bo1**
Tuned	0.2844	**0.3067**	**Del**	0.2956	0.2956	0.2911	0.2889
			Yt	0.2844	0.2956	**0.3067**	0.2844
Crossval	0.2822	0.2956	**Del**	0.2800	0.2800	0.2800	0.2800
			Yt	0.2800	**0.3044**	0.2889	0.3067
			WT2G				
				Jin et al.		**tag-PRF**	
	no PRF	**term-PRF**		**tag-PRF**	**Idf**	**TfIdf**	**Bo1**
Tuned	0.4660	0.5000	**Del**	0.4740	0.4860	0.4740	0.4760
			Yt	0.4780	**0.5040**	0.5000	0.5000
Crossval	0.4660	0.4960	**Del**	0.4720	0.4860	0.4680	0.4700
			Yt	0.4740	**0.5100**	0.5060	0.4980
			BLOG06				
				Jin et al.		**tag-PRF**	
	no PRF	**term-PRF**		**tag-PRF**	**Idf**	**TfIdf**	**Bo1**
Tuned	0.6220	0.6320	**Del**	0.6340	0.6200	0.6200	0.6160
			Yt	0.6500	**0.6760**	0.6520	0.6720
Crossval	0.6220	0.5880	**Del**	0.6320	0.6240	0.6180	0.6160
			Yt	**0.6580**	0.6560	0.6500	0.6360

LAT

			term-PRF +			
		best	**Jin et al.**	**tag-PRF**		
		single PRF	**tag-PRF**	**Idf**	**TfIdf**	**Bo1**
Delicious	**Tuned**	0.3033	0.3068	**0.3080**	0.3073	0.3070
	Crossval	**0.2912**	0.2792	0.2824	0.2750	0.2752
YouTube	**Tuned**	0.3033	0.3042	**0.3163**	0.3134	0.3132
	Crossval	**0.2912**	0.2835	0.2885	0.2843	0.2859

WT2G

			term-PRF +			
		best	**Jin et al.**	**tag-PRF**		
		single PRF	**tag-PRF**	**Idf**	**TfIdf**	**Bo1**
Delicious	**Tuned**	0.3427	0.3439	**0.3453**	0.3452	0.3439
	Crossval	0.3266	**0.3273**	0.3245	0.3264	0.3259
YouTube	**Tuned**	0.3427	0.3450	0.3501	**0.3506**	0.3500
	Crossval	0.3266	0.3245	0.3342	**0.3367**	0.3362

BLOG06

			term-PRF +			
		best	**Jin et al.**	**tag-PRF**		
		single PRF	**tag-PRF**	**Idf**	**TfIdf**	**Bo1**
Delicious	**Tuned**	0.3601	**0.3614**	0.3581	0.3596	0.3596
	Crossval	0.3337	**0.3400**	0.3385	0.3369	0.3381
YouTube	**Tuned**	**0.3760**	0.3645	0.3671	0.3676	0.3670
	Crossval	**0.3723**	0.3434	0.3368	0.3431	0.3425

TABLE 1.4: Best single PRF versus term-PRF + tag-PRF. Performance shown in MAP with Tuned and Cross-validated settings, using tags from Delicious (Del) and YouTube (Yt)

LAT

		best single PRF	term-PRF + Jin et al. tag-PRF	term-PRF + tag-PRF Idf	term-PRF + tag-PRF TfIdf	term-PRF + tag-PRF Bo1
Delicious	**Tuned**	0.3067	**0.3156**	0.3111	0.3133	0.3111
	Crossval	0.2956	**0.3067**	0.2867	0.3000	0.3000
YouTube	**Tuned**	0.3067	0.3133	**0.3200**	**0.3200**	**0.3200**
	Crossval	0.2956	0.2889	0.2956	0.3022	**0.3067**

WT2G

		best single PRF	term-PRF + Jin et al. tag-PRF	term-PRF + tag-PRF Idf	term-PRF + tag-PRF TfIdf	term-PRF + tag-PRF Bo1
Delicious	**Tuned**	**0.5000**	0.4940	0.4860	0.4940	0.4920
	Crossval	0.4960	**0.5040**	0.4920	0.4940	0.4940
YouTube	**Tuned**	0.5000	0.4980	**0.5060**	0.4960	0.4960
	Crossval	0.4960	0.5000	**0.5260**	0.5140	0.5180

BLOG06

		best single PRF	term-PRF + Jin et al. tag-PRF	term-PRF + tag-PRF Idf	term-PRF + tag-PRF TfIdf	term-PRF + tag-PRF Bo1
Delicious	**Tuned**	0.6320	**0.6460**	0.6440	0.6320	0.6340
	Crossval	0.5880	0.5980	0.5980	0.5940	**0.6000**
YouTube	**Tuned**	0.6520	0.6460	**0.6600**	0.6500	0.6440
	Crossval	**0.6500**	0.6020	0.6120	0.6060	0.6000

TABLE 1.5: Best single PRF versus term-PRF + tag-PRF. Performance shown in P@10 with Tuned and Cross-validated settings (for MAP),using tags from Delicious (Del) and YouTube (Yt)

1.4 Conclusions

In this chapter we first surveyed current state-of-the-art approaches for ranking user generated content. As an example application, we proposed a method to use collaborative tags to perform Pseudo Relevance Feedback (PRF). PRF is an IR technique that expands the query with assumed relevant terms and resubmits it for retrieval to the system. We present three different extensions of established term weighting schemes to measure tag salience, different from a previous effort in the same field. We ask whether our proposed tag-PRF approach can enhance performance compared to a standard baseline retrieval model, compared to a competitive term-based PRF model and compared also to an existing tag-based PRF model. A thorough evaluation of the methods on their own and also combined with term-PRF, on three different TREC collections, using two different tag sources and a established retrieval model with tuned and cross-validated settings indicates that our tag-PRF model can enhance retrieval performance consistently, improving the results of an existing Tag-PRF method and rivaling with the established term-PRF method. The proposed Tag-PRF model is especially beneficial when retrieving blogs from the recent BLOG06 collection, which indicates that the lower performance with older collections may be due to the difference in the language of those collections and current language use on the Web.

Given the free availability and increasing popularity (hence amount) of collaborative tagging, further research into incorporating this type of evidence in IR may be fruitful.

Possible next research steps in this direction include investigating the effect of outdated collections upon tag-PRF in a more principled way or how to refine the PRF weighting proposed in this chapter by integrating statistical estimates of the similarity between the query and a tag. Another future research question of interest is to apply tag-PRF to multimedia IR, where the data contains little textual information. In that case, tags could be a way to boost the textual description of the informative content.

Bibliography

[1] Fabian Abel, Ilknur Celik, Geert-Jan Houben, and Patrick Siehndel. Leveraging the semantics of tweets for adaptive faceted search on twitter. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Fridman Noy, and Eva Blomqvist, editors, *International Semantic Web Conference (1)*, volume 7031 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2011.

[2] Fabian Abel, Claudia Hauff, Geert-Jan Houben, Richard Stronkman, and Ke Tao. Semantics + filtering + search = twitcident. exploring information in social web streams. In Ethan V. Munson and Markus Strohmaier, editors, *HT*, pages 285–294. ACM, 2012.

[3] James Allan, Javed A. Aslam, Mark Sanderson, ChengXiang Zhai, and Justin Zobel, editors. *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009.* ACM, 2009.

[4] Gianni Amati. *Probability models for information retrieval based on divergence from randomness.* PhD thesis, University of Glasgow, 2003.

[5] Jaime Arguello, Jonathan L. Elsas, Jamie Callan, and Jaime G. Carbonell. Document representation and query expansion models for blog recommendation. In Eytan Adar, Matthew Hurst, Tim Finin, Natalie S. Glance, Nicolas Nicolov, and Belle L. Tseng, editors, *ICWSM*. The AAAI Press, 2008.

[6] Ricardo A. Baeza-Yates, Arjen P. de Vries, Hugo Zaragoza, Berkant Barla Cambazoglu, Vanessa Murdock, Ronny Lempel, and Fabrizio Silvestri, editors. *Advances in Information Retrieval - 34th European Conference on IR Research, ECIR 2012, Barcelona, Spain, April 1-5, 2012. Proceedings*, volume 7224 of *Lecture Notes in Computer Science*. Springer, 2012.

[7] Bahman Bahmani and Ashish Goel. Partitioned multi-indexing: bringing order to social search. In Mille et al. [44], pages 399–408.

[8] Shenghua Bao, Gui-Rong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. Optimizing web search using social annotations. In Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy, editors, *WWW*, pages 501–510. ACM, 2007.

[9] Niels Olof Bouvin. Unifying strategies for web augmentation. In Jörg Westbomke, Uffe Kock Wiil, John J. Leggett, Klaus Tochtermann, and Jörg M. Haake, editors, *Hypertext*, pages 91–100. ACM, 1999.

[10] Michael Busch, Krishna Gade, Brian Larson, Patrick Lok, Samuel Luckenbill, and Jimmy Lin. Earlybird: Real-time search at twitter. In Anastasios Kementsietsidis and Marcos Antonio Vaz Salles, editors, *ICDE*, pages 1360–1369. IEEE Computer Society, 2012.

[11] Mark James Carman, Mark Baillie, and Fabio Crestani. Tag data and personalized information retrieval. In Ian Soboroff, Eugene Agichtein, and Ravi Kumar, editors, *SSM*, pages 27–34. ACM, 2008.

[12] Mark James Carman, Mark Baillie, Robert Gwadera, and Fabio Crestani. A statistical comparison of tag and query logs. In Allan et al. [3], pages 123–130.

[13] Paul Clough, Colum Foley, Cathal Gurrin, Gareth J. F. Jones, Wessel Kraaij, Hyowon Lee, and Vanessa Murdock, editors. *Advances in Information Retrieval - 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings*, volume 6611 of *Lecture Notes in Computer Science*. Springer, 2011.

[14] Miles Efron. Hashtag retrieval in a microblogging environment. In Fabio Crestani, Stéphane Marchand-Maillet, Hsin-Hsi Chen, Efthimis N. Efthimiadis, and Jacques Savoy, editors, *SIGIR*, pages 787–788. ACM, 2010.

[15] Miles Efron. Information search and retrieval in microblogs. *J. Am. Soc. Inf. Sci. Technol.*, 62(6):996–1008, June 2011.

[16] Miles Efron, Peter Organisciak, and Katrina Fenlon. Improving retrieval of short texts through document expansion. In William R. Hersh, Jamie Callan, Yoelle Maarek, and Mark Sanderson, editors, *SIGIR*, pages 911–920. ACM, 2012.

[17] Miles Efron, Don Turnbull, and Carlos Ovalle. University of Texas School of Information at TREC 2007. In Voorhees and Buckland [66].

[18] Jonathan L. Elsas, Jaime Arguello, Jamie Callan, and Jaime G. Carbonell. Retrieval and feedback models for blog distillation. In Voorhees and Buckland [66].

[19] Jonathan L. Elsas, Jaime Arguello, Jamie Callan, and Jaime G. Carbonell. Retrieval and feedback models for blog feed search. In Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong, editors, *SIGIR*, pages 347–354. ACM, 2008.

[20] Jonathan L. Elsas and Jaime G. Carbonell. It pays to be picky: an evaluation of thread retrieval in online forums. In Allan et al. [3], pages 714–715.

[21] Paul Ferguson, Neil O'Hare, James Lanagan, Owen Phelan, and Kevin McCarthy. An investigation of term weighting approaches for microblog retrieval. In Baeza-Yates et al. [6], pages 552–555.

[22] Dehong Gao, Renxian Zhang, Wenjie Li, Raymond Yiu-Keung Lau, and Kam-Fai Wong. Learning features through feedback for blog distillation. In Wei-Ying Ma, Jian-Yun Nie, Ricardo A. Baeza-Yates, Tat-Seng Chua, and W. Bruce Croft, editors, *SIGIR*, pages 1085–1086. ACM, 2011.

[23] David Hannah, Craig Macdonald, Jie Peng, Ben He, and Iadh Ounis. University of Glasgow at TREC 2007: Experiments in Blog and Enterprise tracks with Terrier. In Voorhees and Buckland [66].

[24] David Hawking and Paul Thomas. Server selection methods in hybrid portal search. In Ricardo A. Baeza-Yates, Nivio Ziviani, Gary Marchionini, Alistair Moffat, and John Tait, editors, *SIGIR*, pages 75–82. ACM, 2005.

[25] Jiyin He, Wouter Weerkamp, Martha Larson, and Maarten de Rijke. An effective coherence measure to determine topical consistency in user-generated content. *Int. J. Doc. Anal. Recognit.*, 12(3):185–203, October 2009.

[26] Lamjed Ben Jabeur, Lynda Tamine, and Mohand Boughanem. Uprising microblogs: a bayesian network retrieval model for tweet search. In Sascha Ossowski and Paola Lecca, editors, *SAC*, pages 943–948. ACM, 2012.

[27] H J Jackson. *Marginalia*. Yale University Press, 2001.

[28] Song Jin, Hongfei Lin, and Sui Su. Query expansion based on folksonomy tag co-occurrence analysis. In *GrC*, pages 300–305. IEEE, 2009.

[29] Mostafa Keikha, Mark James Carman, and Fabio Crestani. Blog distillation using random walks. In Allan et al. [3], pages 638–639.

[30] Mostafa Keikha and Fabio Crestani. Effectiveness of aggregation methods in blog distillation. In Troels Andreasen, Ronald R. Yager, Henrik Bulskov, Henning Christiansen, and Henrik Legind Larsen, editors, *FQAS*, volume 5822 of *Lecture Notes in Computer Science*, pages 157–167. Springer, 2009.

[31] Mostafa Keikha and Fabio Crestani. Linguistic aggregation methods in blog retrieval. *Inf. Process. Manage.*, 48(3):467–475, May 2012.

[32] Mostafa Keikha, Fabio Crestani, and Mark James Carman. Employing document dependency in blog search. *J. Am. Soc. Inf. Sci. Technol.*, 63(2):354–365, February 2012.

[33] Mostafa Keikha, Shima Gerani, and Fabio Crestani. TEMPER: A temporal relevance feedback method. In Clough et al. [13], pages 436–447.

[34] Wai-Lung Lee, Andreas Lommatzsch, and Christian Scheel. Feed distillation using AdaBoost and topic maps. In Voorhees and Buckland [66].

[35] Yeha Lee, Seung-Hoon Na, Jungi Kim, Sang-Hyob Nam, Hun-Young Jung, and Jong-Hyeok Lee. Kle at trec 2008 blog track: Blog post and feed retrieval. In Voorhees and Buckland [67].

[36] Feng Liang, Runwei Qiang, and Jianwu Yang. Exploiting real-time information retrieval in the microblogosphere. In Karim B. Boughida, Barrie Howard, Michael L. Nelson, Herbert Van de Sompel, and Ingeborg Sølvberg, editors, *JCDL*, pages 267–276. ACM, 2012.

[37] Xiaohua Liu, Furu Wei, and Ming Zhou. QuickView: NLP-based tweet search. In *ACL (System Demonstrations)*, pages 13–18. The Association for Computer Linguistics, 2012.

[38] Craig Macdonald and Iadh Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In Philip S. Yu, Vassilis J. Tsotras, Edward A. Fox, and Bing Liu, editors, *CIKM*, pages 387–396. ACM, 2006.

[39] Craig Macdonald and Iadh Ounis. Key blog distillation: ranking aggregates. In Shanahan et al. [59], pages 1043–1052.

[40] Craig Macdonald, Iadh Ounis, and Ian Soboroff. Overview of the TREC 2007 Blog track. In Voorhees and Buckland [66].

[41] Kamran Massoudi, Manos Tsagkias, Maarten de Rijke, and Wouter Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In Clough et al. [13], pages 362–367.

[42] Donald Metzler and W. Bruce Croft. Linear feature-based models for information retrieval. *Inf. Retr.*, 10(3):257–274, June 2007.

[43] Donald Metzler, Congxing Cai, and Eduard H. Hovy. Structured event retrieval over microblog archives. In *HLT-NAACL*, pages 646–655. The Association for Computational Linguistics, 2012.

[44] Alain Mille, Fabien L. Gandon, Jacques Misselis, Michael Rabinovich, and Steffen Staab, editors. *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*. ACM, 2012.

[45] Gilad Mishne. AutoTag: a collaborative approach to automated tag assignment for weblog posts. In Les Carr, David De Roure, Arun Iyengar, Carole A. Goble, and Michael Dahlin, editors, *WWW*, pages 953–954. ACM, 2006.

[46] Gilad Mishne. Using blog properties to improve retrieval. In Natalie S. Glance, Nicolas Nicolov, Eytan Adar, Matthew Hurst, Mark Liberman, and Franco Salvetti, editors, *ICWSM*, 2007.

[47] Gilad Mishne and Maarten de Rijke. A study of blog search. In Mounia Lalmas, Andy MacFarlane, Stefan M. Rüger, Anastasios Tombros, Theodora Tsikrika, and Alexei Yavlinsky, editors, *ECIR*, volume 3936 of *Lecture Notes in Computer Science*, pages 289–301. Springer, 2006.

[48] Sérgio Nunes, Cristina Ribeiro, and Gabriel David. FEUP at TREC 2008 Blog track: Using temporal evidence for ranking and feed distillation. In Voorhees and Buckland [67].

[49] Iadh Ounis, Christina Lioma, Craig Macdonald, and Vassilis Plachouras. Research directions in terrier. *Novatica/UPGRADE Special Issue on Web Information Access, Ricardo Baeza-Yates et al. (Eds), Invited Paper*, 2007.

[50] Iadh Ounis, Craig Macdonald, Maarten de Rijke, Gilad Mishne, and Ian Soboroff. Overview of the TREC 2006 Blog track. In Voorhees and Buckland [65].

[51] Seda Ozmutlu, Amanda Spink, and Huseyin C. Ozmutlu. A day in the life of web searching: an exploratory study. *Inf. Process. Manage.*, 40(2):319–345, March 2004.

[52] Jose San Pedro, Tom Yeh, and Nuria Oliver. Leveraging user comments for aesthetic aware image search reranking. In Mille et al. [44], pages 439–448.

[53] Stephen E. Robertson and Steve Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In W. Bruce Croft and C. J. van Rijsbergen, editors, *SIGIR*, pages 232–241. ACM/Springer, 1994.

[54] J. Rocchio. *Relevance Feedback in Information Retrieval*, pages 313–323. 1971.

[55] Tom Rowlands, David Hawking, and Ramesh Sankaranarayana. New-web search with microblog annotations. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *WWW*, pages 1293–1296. ACM, 2010.

[56] Rodrygo L. T. Santos, Craig Macdonald, Richard McCreadie, Iadh Ounis, and Ian Soboroff. Information retrieval on the blogosphere. *Found. Trends Inf. Retr.*, 6(1):1–125, January 2012.

[57] Kazuhiro Seki, Yoshihiro Kino, Shohei Sato, and Kuniaki Uehara. TREC 2007 Blog track experiments at Kobe university. In Voorhees and Buckland [66].

[58] Jangwon Seo and W. Bruce Croft. Blog site search using resource selection. In Shanahan et al. [59], pages 1053–1062.

[59] James G. Shanahan, Sihem Amer-Yahia, Ioana Manolescu, Yi Zhang, David A. Evans, Aleksander Kolcz, Key-Sun Choi, and Abdur Chowdhury, editors. *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008.* ACM, 2008.

[60] Xin Shuai, Xiaozhong Liu, and Johan Bollen. Improving news ranking by community tweets. In Alain Mille, Fabien L. Gandon, Jacques Misselis, Michael Rabinovich, and Steffen Staab, editors, *WWW (Companion Volume)*, pages 1227–1232. ACM, 2012.

[61] Ian Soboroff, Arjen P. de Vries, and Nick Craswell. Overview of the TREC 2006 Enterprise track. In Voorhees and Buckland [65].

[62] Ke Tao, Fabian Abel, Claudia Hauff, and Geert-Jan Houben. Twinder: A search engine for Twitter streams. In Marco Brambilla, Takehiro Tokuda, and Robert Tolksdorf, editors, *ICWE*, volume 7387 of *Lecture Notes in Computer Science*, pages 153–168. Springer, 2012.

[63] Roelof van Zwol, Vanessa Murdock, Lluis Garcia Pueyo, and Georgina Ramírez. Diversifying image search with user generated content. In Michael S. Lew, Alberto Del Bimbo, and Erwin M. Bakker, editors, *Multimedia Information Retrieval*, pages 67–74. ACM, 2008.

[64] Ellen Voorhees, Donna K. Harman, National Institute of Standards, and Technology (US). *TREC: Experiment and evaluation in information retrieval*, volume 63. MIT press Cambridgeˆ eMA MA, 2005.

[65] Ellen M. Voorhees and Lori P. Buckland, editors. *Proceedings of the Fifteenth Text REtrieval Conference, TREC 2006, Gaithersburg, Maryland, USA, November 14-17, 2006*, volume Special Publication 500-272. National Institute of Standards and Technology (NIST), 2006.

[66] Ellen M. Voorhees and Lori P. Buckland, editors. *Proceedings of The Sixteenth Text REtrieval Conference, TREC 2007, Gaithersburg, Maryland, USA, November 5-9, 2007*, volume Special Publication 500-274. National Institute of Standards and Technology (NIST), 2007.

[67] Ellen M. Voorhees and Lori P. Buckland, editors. *Proceedings of The Seventeenth Text REtrieval Conference, TREC 2008, Gaithersburg, Maryland, USA, November 18-21, 2008*, volume Special Publication 500-277. National Institute of Standards and Technology (NIST), 2008.

[68] Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. Unified relevance models for rating prediction in collaborative filtering. *ACM Trans. Inf. Syst.*, 26(3):16:1–16:42, June 2008.

[69] Wouter Weerkamp. *Finding People and their Utterances in Social Media.* PhD thesis, University of Amsterdam, 2011.

[70] Wouter Weerkamp, Krisztian Balog, and Maarten de Rijke. Finding key bloggers, one post at a time. In Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, and Nikolaos M. Avouris, editors, *ECAI*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 318–322. IOS Press, 2008.

[71] Wouter Weerkamp and Maarten de Rijke. Credibility improves topical blog post retrieval. In Kathleen McKeown, Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, editors, *ACL*, pages 923–931. The Association for Computer Linguistics, 2008.

[72] Stewart Whiting, Iraklis A. Klampanos, and Joemon M. Jose. Temporal pseudo-relevance feedback in microblog retrieval. In Baeza-Yates et al. [6], pages 522–526.

[73] Ding Zhou, Jiang Bian, Shuyi Zheng, Hongyuan Zha, and C. Lee Giles. Exploring social annotations for information retrieval. In Jinpeng Huai, Robin Chen, Hsiao-Wuen Hon, Yunhao Liu, Wei-Ying Ma, Andrew Tomkins, and Xiaodong Zhang, editors, *WWW*, pages 715–724. ACM, 2008.