Finding people and their utterances in social media

Weerkamp, W.

Citation for published version (APA):
This chapter contains an overview of previous work related to the topics discussed in this thesis. This related work is presented in six sections and follows the structure of the thesis. We start with a general introduction to information retrieval in Section 2.1, followed by a review section for each research chapter.

**Section 2.2** (Chapter 4) Work related to query log analysis, with a focus on different types of queries, sessions, and users.

**Section 2.3** (Chapter 5) Previous research on blogger finding, blog feed search, and previous applications of techniques we will use.

**Section 2.4** (Chapter 6) Work in the field of (automatic) credibility assessment, both in general web settings and in social media.

**Section 2.5** (Chapter 7) Related work in query modeling in general and external query expansion in particular.

**Section 2.6** (Chapter 8) Literature regarding access to information in email archives and specifically email search.

### 2.1 Information Retrieval

Information Retrieval (IR) deals with the representation, storage, organization of, and access to information items [11]. Generally speaking we can divide IR in two processes: (i) indexing and (ii) searching. The first process focuses on representation, storage, and organization, while the second process concerns access to the information items, usually in response to an information need. Search approaches, or retrieval models, can be classified into several main classes: Boolean models, vector space models, and probabilistic models. In this section we briefly discuss each of the approaches and how they differ from each other.

The (original) *Boolean model* is a set-based retrieval model using Boolean algebra, which allows users to translate their information need into queries containing AND, OR, and NOT operators. The AND operator places all terms in a conjunction (i.e., documents should contain all query terms), whereas the OR operator places them in a disjunction (i.e., documents should contain any of the query terms). The NOT operator dictates
2. Background

which terms are indicative of irrelevant documents. Given a Boolean query, the model returns a set of (potentially) relevant documents. The decision of the relevancy of a document is a binary one, a document is either relevant and therefore included in the set of retrieved documents or not relevant and is thus ignored. This binary decision also prevents Boolean models from ranking the documents in the retrieved set, as they are all considered equally relevant. Joyce and Needham [86], however, proposed to use a term frequency-related technique to rank documents in a Boolean retrieval system.

Although the Boolean model is relatively easy to understand, it cannot deal with partially relevant documents. Besides that, sets of retrieved documents quickly turn too small (in case of too restrictive queries) or too large (in case of too general queries). The next-generation models, i.e., vector space models, therefore allowed for partial matching of documents and queries, leading to a ranking of documents based on how well they matched to the query. The vector space model [162, 163] does allow for partial matching of query and document; it places both the document and the query as vectors in a vector space, where the dimensions are defined by the vocabulary. The similarity between document and query is consequently measured, for example, by the cosine of the angle between the two vectors (i.e., cosine similarity). Components of the vectors can take binary, as well as real values. In case of the latter, Salton and McGill [164] presented various options to weight terms. The most commonly used weights are term frequency (TF), that is, the relative frequency of a term in a document, and the inverse document frequency (IDF), which indicates how useful a term is for distinguishing between documents. The vector space model allows for partial matching and generates a ranking based on the similarity between a document and the query. Even more so, the simplicity of the model makes it very efficient without losing effectiveness, making the vector space model the leading retrieval model for many years.

The third class of retrieval models are the probabilistic models. Robertson and Spärck Jones [154] took the notion of relevance from Maron and Kuhns [127] and developed the probability ranking principle (PRP). Here, the probability of a document being relevant to the user’s query is estimated. The initial model is often referred to as the binary independence retrieval model, because it explicitly contains the probability of a document being relevant and the probability of the same document not being relevant [153, 154]. The success of this retrieval model depends on the availability of the distributions of terms over relevant and non-relevant documents and these distributions are usually unknown. The initial model uses binary weights for query terms in documents, which was later changed by Robertson et al. [156] to include term frequencies.

One of the most used retrieval models is Okapi BM25 [178]. The Okapi system is based on PRP, but after its initial failure in TREC-1 [61], Robertson and Walker [155] explored other weighting schemes, taking into account document length and term frequency. These experiments led to BM25, which is still a very competitive system and a hard-to-beat baseline in many IR research papers.

A retrieval approach that gained momentum over the last couple of years is the learning to rank approach [112]. As the name suggests, learning to rank is based on machine learning techniques and given the amount of training data that is available nowadays, it is feasible to apply machine learning techniques to the problem of ranking documents. Learning to rank tries to learn the best way of combining features extracted from query-document pairs, like query term frequency, document length, number of inlinks, etc. The
rationale behind using learning to rank is that the number of features we can use to rank documents becomes too big for anything else than a machine learning approach. Although learning to rank is an interesting retrieval framework that has shown promising results, we consider it beyond the scope of this thesis.

Language modeling for information retrieval

In this thesis we use language modeling for IR as our retrieval model. A statistical language model is simply a probability distribution over all possible units [159], where a unit can be anything, ranging from documents to sentences (as is the case in the following example). Statistical language models gained popularity in the 1970’s in the setting of automatic speech recognition [81]. In that setting, the goal is to find the sentence \(s\) that is most likely to have been spoken in a given an acoustic signal \(a\):

\[
s^* = \arg \max_s P(s|a) = \arg \max_s P(a|s) \cdot P(s),
\]

(2.1)

where \(P(s)\) is the language model. Sentence \(s\) is observed as having been generated by some probability and transmitted through a noisy channel that transforms \(s\) to signal \(a\) with probability \(P(a|s)\). Using this model we are not limited to selecting one sentence \(s\), but we can rank various sentences according to their probability. We find that this characteristic is useful in IR too.

The first suggestion to use language models in information retrieval came from Ponte and Croft [149]. This work was soon followed by work from Hiemstra [71] and Miller et al. [133], who both use a (simple) multinomial language model. This model is still the most commonly used application of language models for IR. Both BM25 and language modeling are now often used as baselines against which new retrieval models are compared or on top of which new techniques are applied.

We also use language modeling as our baseline retrieval model on top of which we apply blogger finding models (Chapter 5), credibility indicators (Chapters 6 and 8), and external query modeling (Chapters 7 and 8). More details on the language modeling approach can be found in Section 3.3, in which we introduce the baseline retrieval model for this thesis.

We have given a brief introduction to the main classes of retrieval models. Many more flavors of retrieval models exist, but it is beyond the scope of this thesis to list all of these. Instead, we refer to textbooks by Baeza-Yates and Ribeiro-Neto [11] and Manning et al. [126], who both give thorough reviews of a large number of retrieval models and other techniques related to information retrieval (e.g., indexing, query expansion, . . .). We continue our literature review with work related to query log analysis, which is the topic of Chapter 4.

2.2 Query Log Analysis

One of the first large scale query log analysis papers uses search logs of AltaVista [174]. The authors perform a descriptive analysis of the (almost) 1 billion queries in the log,
indicating query length (mostly 1–3 term queries), session length (mostly one query session), popular query terms (sex related), the number of result pages a user looks at (mostly one page), and how queries are modified within a session. Following several other studies of web search engine logs, Jansen and Spink [78] compare nine search engine logs created between 1997 and 2002. They conclude that most findings are stable over time, but that, e.g., the percentage of users who only look at the first result page increases. They also show that the percentage of queries related to people, places or things (“entities”) increases from 21% in 2001 to over 41% in 2002, clearly indicating the importance of people search.

When it comes to people search and query log analysis, not much work has been done. Guo et al. [60] propose a method to recognize named entities in queries by learning context for these entities. Although their work shows promise, it focuses on entities like books, movies and music, rather than people. More closely related work is done by Pound et al. [150] and looks at ad-hoc object retrieval; the authors show that over 40% of queries in their dataset are of type “entity” and they specify methods for dealing with such queries in a “web of data” setting.

2.2.1 Queries

What is it that people are searching for in a particular search environment? This question is the rationale behind many papers covering queries and query types. Classification of queries is often based on (i) query intent or (ii) query semantics. An influential paper of the former type by Broder [26] looks at queries in a web search engine. An exploration of query log data reveals three types of query: informational, navigational, and transactional. Most queries in a web search engine are informational (40–50%), followed by transactional (30–36%). Later work by Rose and Levinson [158] extends this taxonomy with subclasses. A manual classification of 1,500 web queries shows that the percentage of informational queries is higher than in the original paper (about 60%), at the cost of both other types.

The rise of verticals leads to users interacting with specialized search systems, which in turn might lead to different types of queries and different search behaviors. Mishne and de Rijke [138] acknowledge this and look at query types in a blog search engine. Since almost all blog queries are informational they propose two new query types: concept and context queries—both of which are informational but quite distinctive in blog search. Another type of vertical search that is explored using query logs are audiovisual archives [75]. Here, the authors do not classify queries, but show general statistics of the logs, indicating that users mainly look for program titles and entities (organizations, people). These two papers show that, by moving towards more specialized search engines, the query typology needs refinement too.

Looking at query classification research based on query semantics, there exists a large body of related work that considers queries that a given query co-occurs with (see “Sessions”). One example is the classification of query refinements, addressed in [74]. A different classification task is proposed by Cao et al. [31], who state that query context (i.e., previous queries in the same session) is needed to classify queries into categories.
2.2. Query Log Analysis

2.2.2 Sessions

Sessions are an important aspect in query log analysis, and various ways of detecting sessions have been proposed. According to Jansen [77], session duration is the interval between the user submitting the first query and the user “leaving” the search engine, resulting in sessions varying from several seconds to a few hours. Most time-based session detection approaches group logged actions by some user id, sort the actions chronologically for each user, and split sessions on intervals longer than a certain cutoff value. The choice of cutoff value is dependent on the goal of the analysis. For example, based on a manual examination Mishne and de Rijke [138] use very small cutoff values between 10 and 30 seconds and show that these values mimic sessions based on query reformulation. Longer sessions (e.g., 30 minutes [79]) allow one to explore the different queries and query types a user issues.

Although the time-based approach is a commonly used definition of sessions, there are alternatives. Huang and Efthimiadis [74] use query reformulations to identify session boundaries. Here, sessions consist of consecutive queries by the same user, where each query is a reformulation of the previous query (e.g., adding or deleting words). The idea is that all reformulated queries address a single underlying information need and should be in one session. Jansen et al. [79] compare query reformulations for session detection to the time-based detection; they conclude that query reformulation results in more detected sessions.

A different approach has been proposed by Lucchese et al. [115], who try to detect sessions based on a user’s task. Since multitasking is very common in web search, they conclude that time-based techniques fail at task-dependent session detection; instead, they propose to cluster queries and use the clusters for session detection.

2.2.3 Users

Research into user behavior from query logs can be challenging, since it can be hard to determine which queries and sessions belong to the same user. White and Drucker [207] counter this issue by using a set of volunteer users. They collect search data from these users over a five month period. From this data, they identify two user types: navigators (users with consistent search behavior) and explorers (variable behavior). A different approach (in the setting of searching literature in CiteSeer) by Manavoglu et al. [124] tries to model user behavior and predicts actions by similar users, based on previous users’ actions.

Where the two studies just mentioned model users based on their actions, Weber and Jaimes [193] describe users’ demographics. For this, they use characteristics per ZIP code, and election results per county. Combining demographics with what people are searching for and how they do so, allows them to gain insight in the behavior of users with specific characteristics.

In Chapter 4 we analyze a query log of a people search engine and explore each of the three information objects mentioned above (i.e., queries, sessions, and users) in detail.
2. Background

2.3 Blogger Finding

Some commercial blog search facilities provide an integrated blog search tool to allow users to easily find new blogs of interest. In [57], a multi-faceted blog search engine was proposed that allows users to search for blogs and posts. One of the options was to use a blogger filter: the search results (blog posts) are clustered by blog and the user is presented with a list of blogs that contain one or more relevant posts. Ranking of the blogs is done based on the EigenRumor algorithm [56]; in contrast to the methods that we consider below, this algorithm is query-independent.

An important theme to emerge from the work on systems participating in the TREC 2007 and 2008 blog feed search tasks is the indexing unit used [119]. While the unit of retrieval is fixed for blog feed search—systems have to return blogs in response to a query—it is up to the individual systems to decide whether to produce a ranking based on a blog index or on a post index. The former views blogs as a single document, disregarding the fact that a blog is constructed from multiple posts. The latter takes samples of posts from blogs and combines the relevance scores of these posts into a single blog score. The most effective approaches to feed distillation at TREC 2007 were based on using the (aggregated) text of entire blogs as indexing units. E.g., Elsas et al. [49, 51] 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 best performance in terms of MAP [6]. Although the large document approach is competitive in terms of performance, it is considered unrealistic by most researchers, leaving the small document approaches as the way to go.

Participants in TREC 2007 and 2008 [120] explored various techniques for improving effectiveness on the blog feed search task: Query expansion using Wikipedia [49], topic maps [108], and a particularly interesting approach—one that tries to capture the recurrence patterns of a blog—using the notion of time and relevance [167]. Although some of the techniques used proved to be useful in both years (e.g., query expansion), most approaches did not lead to significant improvements over a baseline, or even led to a decrease in performance, proving the challenging nature of the task.

Other approaches that were applied to this task are the use random walks [92], where connections between blogs, posts, and terms are considered. Although time is an important aspect in blogs, it is often ignored. Keikha et al. [93] propose a method that does take time into account and use time-dependent representations of queries and blogs to measure the recurring interest of blogs.

In the setting of blog feed search, authors have considered various ways of improving effectiveness: (i) index pruning techniques, (ii) modeling topical noise in blogs to measure recurring interest, (iii) using blog characteristics such as the number of comments, post length, or the posting time, (iv) mixing different document representations, and (v) sampling posts for score aggregation. We briefly sample from publications on each of these four themes.

Starting with index pruning, a pre-processing step in [169] consists of removing all blogs that consist of only one post, since retrieving these blogs would come down to retrieving posts and would ignore the requirement of retrieving blogs with a recurring in-
terest. We use various types of index pruning in Section 5.3 and 5.4, including removing non-English blogs and blogs that consist of a single post.

As to capturing the central interest of a blog, several authors attempt to capture the central interest of a blogger by exploiting information about topical patterns in blogs. The voting-model-based approach of [117] is competitive with the TREC 2007 blog feed search results reported in [119] and formulates three possible topical patterns along with models that encode each into the blog retrieval model. In [66] the need to target individual topical patterns and to tune multiple topical-pattern-based scores is eliminated; their proposed use of a coherence score to encode the topical structure of blogs allows them to simultaneously capture the topical focus at the blog level and the tightness of the relatedness of sub-topics within the blog. A different approach is proposed in [168], where the authors use diversity penalties: blogs with a diverse set of posts receive a penalty. This penalty is integrated in various resource selection models, where a blog is seen as a resource (collection of posts), and given a query, the goal is to determine the best resource. Below, we capture the central interest of a blogger using the KL-divergence between a post and the blog to which it belongs.

The usage of blog-specific features like comments and recency has been shown to be beneficial in blog post retrieval [136, 194]. In blog feed search these features can be applied in the post retrieval stage of the Posting model, but they can also be used to estimate the importance of a post for its parent blog [197]; we use some of these features in Section 5.3 and 5.4.

Finally, blog posts can be represented in different ways. On several occasions people have experimented with using syndicated content (i.e., RSS or ATOM feeds) instead of permalinks (HTML content) [49, 51, 136]; 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; combinations of various representations show increased effectiveness in other web retrieval tasks (e.g., ad hoc retrieval [48, 84]). We increase the efficiency of our most effective model by considering multiple content representations in Section 5.4.

Elsas and Carbonell [50] apply their large and small document models to forum thread retrieval and find that small document models work better, especially when only a sample of relevant forum posts is used. A similar conclusion is drawn by Keikha and Crestani [91], who explore the effects of various aggregation methods on blog feed search and find that taking only the top relevant posts in a blog leads improvements over a baseline in which all posts are considered when aggregating scores. These post selection techniques are applied after the relevance of posts has been determined. In Chapter 5 we select posts before determining relevance.

## 2.4 Credibility in Web Settings

In a web setting, credibility is often couched in terms of authoritativeness and estimated by exploiting the hyperlink structure. Two well-known examples of algorithms that do this are the PageRank and HITS algorithms [111], that use the link structure in a topic independent or topic dependent way, respectively. The idea behind these algorithms is that more pages linking to a certain document is an indication of this page being more au-
2. Background

2.4.1 Credibility in social media

Credibility-related work in social media comes in various forms, and is applied to different platforms. Weimer et al. [205] discuss the automatic assessment of forum post quality; they use surface, lexical, syntactic and forum-specific features to classify forum posts as bad posts or good posts. The use of forum-specific features (such as whether or not the post contains HTML, and the fraction of characters that are inside quotes of other posts), gives the highest benefits to the classification.

Working in the community question/answering domain, Agichtein et al. [3] use content features, as well non-content information available, such as links between items and explicit quality ratings from members of the community to identify high-quality content. In the same domain, Su et al. [183] try to detect text trustworthiness by incorporating evidentiality (e.g., “I’m certain of this”) in their feature set.

To allow for better presentation of online reviews to users, O’Mahony and Smyth [143] try to determine the helpfulness of reviews. Their features are divided in reputation features, content features, social features, and sentiment features. Follow-up work also includes readability features [144].

For blogs, most work related to credibility is aimed at trying to identify blogs worth following. Sriphaew et al. [179] try to identify “cool blogs,” i.e., blogs that are worth exploring. Their approach follows a combination of credibility-like features with topic consistency, as used in blog feed search [119]. Similar work is done by Chen and Ohta [35], who try to filter blog posts using topic concentration and topic variety. The impact of post length was further explored by Hearst and Dumais [67]. They found that there is a correlation between the length of posts in a blog and the popularity of that blog. Mishne and de Rijke [138]’s observation that bloggers often report on news events is the basis for the credibility assessment in [87]. The authors compare blog posts to news articles about the same topic, and assign a credibility level based on the similarity between the two. In Chapter 6 we use a similar technique, but acknowledge that not all blog posts are about news events, hence the need for other indicators. Spam identification may be part of estimating credibility, not only for blogs (or blog posts), but also for other (web) documents. Spam identification has been successfully applied in the blogosphere to improve retrieval effectiveness, for example in [80, 136].

Recently, credibility indicators have been successfully applied to post finding in a specific type of blog environment: microblogs [128]. Besides translating indicators to the new environment, the authors also introduced platform-specific indicators like followers, retweets, and recency. For the task of exploring trending topics on Twitter, Castillo et al. [33] use a similar set of indicators to assess credibility of tweets, and use human
2.5. Query Modeling

To bridge the vocabulary gap between the query and the document collection we often use query modeling. Query modeling consists of transformations of simple keyword queries into more detailed representations of the user’s information need, for example by assigning (different) weights to terms, expanding the query with terms related to the query, or using phrases. Many query expansion techniques have been proposed and they mostly fall into two categories, i.e., global analysis and local analysis. The idea of global analysis is to expand the query using global collection statistics based, for instance, on a co-occurrence analysis of the entire collection. Thesaurus- and dictionary-based expansion as, e.g., in [151], also provide examples of the global approach.

Our focus is on local approaches to query expansion, that use the top retrieved documents as examples from which to select terms to improve the retrieval performance [157]. In the setting of language modeling approaches to query expansion, the local analysis idea has been instantiated by estimating additional query language models [103, 184] or relevance models [105] from a set of feedback documents. Yan and Hauptmann [210] explore query expansion in a multimedia setting. Meij et al. [130] introduce a model that does not depend solely on each feedback document individually nor on the set of feedback documents as a whole, but combines the two approaches. Balog et al. [17] compare methods for sampling expansion terms to support query-dependent and query-independent query expansion; the latter is motivated by the wish to increase “aspect recall” and attempts to uncover aspects of the information need not captured by the query. Kurland et al. [101] also try to uncover multiple aspects of a query and to that end they provide an iterative “pseudo-query” generation technique, using cluster-based language models.

2.5.1 External query expansion

The use of external collections for query expansion has a long history, see, e.g., [102, 161]. Diaz and Metzler [44] were the first to give a systematic account of query expansion using an external corpus in a language modeling setting, with the goal of improving the
2. Background

estimation of relevance models. As will become clear in Section 7.1, Diaz and Metzler [44]'s approach is an instantiation of our general model for external expansion.

Typical query expansion techniques, such as pseudo-relevance feedback, using a blog or blog post corpus do not provide significant performance improvements and often dramatically hurt performance. For this reason, query expansion using external corpora has been a popular technique at the TREC Blog track [146]. For blog post retrieval, several TREC participants have experimented with expansion against external corpora, usually a news corpus, Wikipedia, the web, or a mixture of these [54, 80, 216]. For the blog finding task introduced in 2007, TREC participants again used expansion against an external corpus, usually Wikipedia [6, 19, 49, 54, 55]. The motivation underlying most of these approaches is to improve the estimation of the query representation, often trying to make up for the unedited nature of the corpus from which posts or blogs need to be retrieved. Elsas et al. [51] go a step further and develop an interesting query expansion technique using the links in Wikipedia.

Another approach to using external evidence for query expansion is explored by Yin et al. [212]. They use evidence found in web search snippets, query logs, and web search documents to expand the original query and show that especially the snippets (generated by web search engines) are very useful for this type of query expansion. Xu et al. [208] apply query expansion on Wikipedia after classifying queries into entity, ambiguous, and broader queries and find that this external expansion works well on various TREC collections. This work shows some resemblance to our work in Chapter 7, but it also shows large differences. The method proposed by Xu et al. [208] is a two-step approach and makes a binary decision how to expand the query. Our model is a one-step approach and is more general in that it can mix various external collections based on the query without making a binary decision of whether or not to expand the query on a certain collection. Our work in Chapter 7 shows more resemblance to the mixture of relevance models of Diaz and Metzler [44], which is in fact one of the instances of our general query expansion model.

2.6 Email Search

Research on access to collections of email messages has traditionally focused on tools for managing personal collections, in part because large and diverse collections were not available for research use [52]. Triggered by the introduction of the Enron [96] and W3C [192] collections, opportunities opened up to study new challenges. A large body of these efforts focused on people-related tasks, including name recognition and reference resolution [45, 53, 134, 135], contact information extraction [13, 42], identity modeling and resolution [52], discovery of peoples’ roles [109], and finding experts [13, 166, 214]. The Enron email collection is a popular resource within the e-discovery community. TREC Legal [37, 69] has been using this collection since 2009 to answer research questions related to finding responsive documents for a given production request. Another line of work centers around efficient access to email-based discussion lists. Tuulos et al. [190] introduce a system that provides access to large-scale email archives from multiple viewpoints, using faceted search. Newman [142] explores visualization techniques to aid the coherent reading of email threads. Following this line of work, a number of research
2.6. Email Search

groups explored email search as part of the TREC 2005 [38] and 2006 [177] Enterprise tracks. Common approaches include the use of thread information to do document expansion, the use of filters to eliminate non-emails from the collection, assigning different weights to fields in emails (ads, greetings, quotes, etc), and smoothing the document model with a thread model.

One can view email as user-generated content: after subscribing to a mailing list, users are free to send whatever they want to the list, without an editor stopping them. Communicating through a mailing list is, in a way, comparable to blogging: it is one-to-many communication, readers have the possibility to respond (email or comments), there are no rules on what to write, and both have a similar structure (blog-posts-comments vs. thread-mails-quotes). Much of the work presented in previous sections is therefore applicable to email finding (e.g., credibility, quality, and (external) query expansion). In an early paper, Lewis and Knowles [110] identify the need for threading of emails and they show that they can retrieve the parent email of a reply successfully using the quoted text as a query. In our case, threads are given, but we use the fact that emails in the same thread share content to our advantage. Seo et al. [171], in a follow-up on [170], propose retrieval methods for communities, like mailing lists, that make use of hierarchical structures. They investigate how to detect threads automatically and find that using these thread structures in retrieval can lead to significant improvements. Very similar work is done by Duan and Zhai [46] who use smoothing techniques based on thread structures for forum post retrieval. These lines of work are related to our approaches in Chapter 8, where we apply query expansion based on different context levels to improve email search. The main difference between previous work and the work in Chapter 8 is that we not only look at threads, but also explore larger context levels like the whole mailing list and the community website. Besides that, we also explore the use of credibility-inspired indicators (viz. Chapter 6) on top of each of the context levels.