Learning to Rank Academic Experts

Catarina Moreira

Instituto Superior Técnico, INESC-ID
Av. Professor Cavaco Silva, 2744-016 Porto Salvo, Portugal

Abstract. The task of expert finding has been getting increasing attention in the information retrieval literature. Still, the current state-of-the-art lacks in principled approaches for combining different sources of evidence in an optimal way. In the context of my MSc thesis, I will explore the usage of learning to rank methods as a principled approach for combining multiple estimators of expertise. This paper surveys the current state-of-the-art in both expert finding and learning to rank. It presents the fundamental concepts and the most important related works, also detailing my thesis proposal and the envisioned validation plan.

1 Introduction

The automatic search for knowledgeable people in the scope of specific communities or large organizations, with basis on documents describing people's activities, is an information retrieval problem that has been receiving increasing attention (Serdyukov, 2009). Usually referred to as *expert finding*, the task involves taking a short user query as input and returning a list of people sorted by their level of expertise in what concerns the query topic.

Early experiments in the area started in the late 90s with the pioneering work of Mattox et al. (1999). They applied information retrieval, information extraction and collaborative filtering techniques to perform the search. The ranking of the candidates was computed by the number of mentions of the candidate names associated with the query terms in the documents. The search was performed over newsletters and publications. They considered that a candidate is an expert if it is linked to a wide range of documents and/or a large number of documents containing the query terms.

An expert finding task also appeared at the Text REtrieval Conference (TREC) in year of 2005, due to the increasing interest in this area. TREC provided a platform where researchers could test their techniques through a scenario that included the discovery of relationships between entities in a large organizational scale. The first approaches that were tested involved creating a document for each candidate and then applying simple IR techniques to rank those documents, or involved natural language processing and information extraction technologies (Craswell et al., 2005).

Since TREC, many effective expert finding approaches have been proposed in the literature, exploring different retrieval models and different sources of evidence for estimating the candidate's expertise. However, the current state-of-the-art still lacks in principled approaches for combining different sources of evidence in an optimal way.

In traditional information retrieval tasks such as ad-hoc retrieval, there has been an increasing interest on the usage of machine learning methods for building retrieval formulas capable of estimating relevance for query-document pairs (Liu, 2009). The general idea is to use hand-labeled data (e.g., document collections containing relevance judgments for specific sets of queries, or information regarding user-clicks aggregated over query logs) to train ranking models, this way leveraging on data to combine the different estimators of relevance in an optimal way. However, few previous works have specifically addressed the usage of learning to rank approaches in the task of expert finding.

In the scope of my MSc thesis, I will explore the usage of learning to rank methods in the expert finding task, specifically combining a large pool or estimators for relevance/expertise (e.g., estimators derived from textual similarity, from the graph-structure of the community of experts, and from profile information about the experts). This document presents a survey on the information retrieval subjects of expert finding and learning to rank, detailing the fundamental concepts that will be used throughout my work and also presenting my thesis proposal and the envisioned validation plan.

The rest of this document is organized as follows: Section 2 presents the main concepts used throughout the document. Section 3 details previous works specifically handling the problem of expert finding, through heuristic models. Section 4 details previous works that address the expert finding problem through learning to rank methods. Section 5 introduces several features upon which one can leverage for estimating relevance in an expert finding system. Section 6 presents my thesis proposal and outlines the envisioned validation plan, specifically detailing the datasets and evaluation metrics that will be used. Finally, Section 7 summarizes the most important aspects.

2 Concepts

This section presents important concepts which are critical for understanding this work. It gives a brief explanation of what is information retrieval and introduces the task of expert search. It also presents the classic models used in information retrieval, as well as recent advancements such as linked-based approaches. Finally, the most popular metrics for evaluating information retrieval systems are presented.

2.1 Information Retrieval and Expert Search

Information Retrieval (IR) aims at the representation, organization, storage and retrieval of documents, in order to provide the user with easy access to required information (Baeza-Yates and Ribeiro-Neto, 1999). To satisfy an information need, an IR system must be able to check the documents in a collection and rank them according to their degree of relevance to a user query expressing the information need. The major difficulty faced by an IR system resides in the computation of these relevance estimates.

Expertise retrieval is a subcategory of information retrieval where the goal is to identify a list of expert candidates who are knowledgeable about a given expertise area, by uncovering associations between those candidates and topics discussed over a document collection (Balog et al., 2007). A user starts by formulating a query with a topic of his interest. Then, the retrieval system ranks the candidates according to the expertise area expressed in the query, using available documentary evidence and profile information related to the candidate experts (Serdyukov, 2009).

In order to work, an expert search system has two requirements, namely (i) a list of candidates that can be retrieved by the system, and (ii) textual evidence on the expertise of the candidates, such as e-mails, academic publications, blogs, web pages, forum posts, etc. (Macdonald and Ounis, 2006).

Expert search is much more difficult than document retrieval, because (i) the expertise areas of a candidate are rare and hard to quantify and (ii) the experience of the candidates may vary (Maybury, 2006). According to Macdonald et al. (2008), the three major factors which affect the retrieval performance of an expert search system are (i) the selection of the documents associated with the candidates, (ii) the measurement of the relevance for each document which contains the query topics, and (iii) the combination of the expertise evidence from the associated documents.

A standard retrieval system cannot solve the expert search problem directly, since it would find experts strictly by ranking documents (Petkova and Croft, 2006). The system may start by retrieving documents, but it must then extract and process the textual information to uncover the associations between documents and topics and documents and candidates. There are two main approaches for modeling these associations. The first one is based on gathering all the information from the different candidates in profile documents, and then rank the candidates with basis on the associations found between the topic and the query. The other approach first collects all documents that are associated with the topic and then tries to find associations to the candidate experts. In both these approaches, the expert finding system has to discover documents related to a person and then estimate the probability of that person being an expert with basis on the text. Section 3 of this document provides an exhaustive survey on the subject of expert finding.

2.2 Classical Models of Information Retrieval

In the context of ad-hoc document retrieval, several conceptual models have been used to represent documents and queries. The classical models of information retrieval can be classified in three major categories, namely Vector Space Models, Discriminative Probabilistic Models and Generative Probabilistic Models (i.e., Language Models).

2.2.1 Vector Space Models

In this case, documents are represented as vectors in an n-dimensional vector space, where n corresponds to the number of unique terms in the document collection and where each term of the vector can be weighted according to its relative importance. Each document i is expressed as $D_i = (d_{i1}, d_{i2}, ..., d_{in})$, where d_{ij} corresponds to the weight of the d_{jth} term in the document i.

Having both documents and queries represented in an n-dimensional space, one possible and very effective way to measure the similarity between them is through vector matching operations such as the cosine similarity. This operation is used to measure the cosine of the angle between the vectors. The similarity results are then used to rank the documents according to their estimated relevance to the query (Salton et al., 1975).

In order to score the documents correctly according to their importance to the query, it is critical to know which weights should be applied to each unique term of the vectors. Two of the most effective term weighting functions are the TF.IDF and the Okapi BM25 approaches. BM25 was originally proposed as a scoring function which determines the similarity of two documents (i.e, it offers a complete retrieval model). However, Okapi BM25 can also be used as a term weighting function.

TF.IDF is based on calculating the relative frequency of words in a specific document, compared to the inverse proportion of that word over the entire document corpus (Ramos, 2001). The TF.IDF formula is given by:

$$TF.IDF(d) = f_{t,d} \times \log \frac{|D|}{f_{t,D}} \tag{1}$$

where $f_{t,d}$ is the number of occurrences of term t in document d, $|D|$ is the size of the document collection and $f_{t,D}$ corresponds to the number of documents in the collection where term t occurs.

According to Manning (2008), the TF.IDF values are high when the term t occurs many times in a small set of documents and are low when the term t occurs a few times in the document or when t occurs very often in a large number of documents (e.g., terms like articles and prepositions).

The Okapi BM25 term weighting and document-scoring function (Robertson and Zaragoza, 2009) is a combined function composed of several simpler scoring functions with different components and parameters. The components which are involved in BM25 are mainly the *inverse document frequency* and term *term frequency*, which have been explained in the context of the TF.IDF metric, together with the *document length* and the *average document length* of the collection. The BM25 document-scoring function is given by Equation 2, where $i \in \mathit{Terms}(q)$ represents the set of terms from query q, $\mathit{Freq}(i, d)$ is the number of occurrences of term i in document d, $|d|$ is the number of terms in document d, and $\mathcal{A}$ is the average length of the documents in the collection.

$$BM25(q,d) = \sum_{i \in Terms(q)} \log\left(\frac{N - Freq(i) + 0.5}{Freq(i) + 0.5}\right) \times \frac{(k_1 + 1) \times \frac{Freq(i,d)}{|d|}}{\frac{Freq(i,d)}{|d|} + k_1 \times (1 - b + b \times \frac{|d|}{\mathcal{A}})} \tag{2}$$

The function in Equation 2 measures the similarity of a query q towards a given document d in a collection by summing the weights of all the terms of the query. The results can be used to rank documents according to their relevance to the query. If we remove the $\sum_{i \in Terms(q)}$ summation from Equation 2 and make the formula independent of a query parameter, then Okapi BM25 can be used as a term scoring function.

One limitation of Okapi BM25 is that it does not provide any guidelines on how to choose the internal parameters k_1 and b. However, according to Robertson and Zaragoza (2009), these parameters provide good results when their values are between $0.5 < b < 0.8$ and $1.2 < k_1 < 2$.

2.2.2 Discriminative Probabilistic Models

In the case of discriminative probabilistic models, the retrieval system tries to estimate the probability that a specific document d_m is to be judged relevant or not relevant with respect to a user query q_k, *i.e*, $P(R|q_k, d_m)$. This is often referred to as the probability ranking principle, stating that the ranking of the documents is given by their estimated probability of relevance with respect to the query $P(R = 1|d, q)$. To estimate this probability, it is assumed that terms are distributed differently within relevant and non-relevant documents. Let $T = \{t_1, ..., t_n\}$ denote the set of terms in the collection. The set of terms which occur in document d_m^T can be represented as a binary vector $x = (x_1, ..., x_n)$ where $x_i = 1$ if $t_i \in d_m^T$ and $x_i = 0$ otherwise (Fuhr, 1992). Instead of these binary vectors, weighting functions such as TF.IDF can also be used to improve the retrieval mechanism. The probability $P(R|q_k, x)$ is estimated through a particular discriminative model and different documents containing the same set of terms are to be ranked with the same probability. These estimates are important to evaluate how terms in a document contribute to the relevance judgment. With this information, the documents are then ordered by decreasing estimated probability of relevance.

The original and still most influential discriminative probabilistic retrieval model is the *binary independence model* (BIM). The Binary Independence Model uses the Naive Bayes probabilistic theories and it is based on the independence assumption, i.e. it assumes that terms occur in the documents independently. Although this independence approach is not very correct, studies have shown that in practice it usually gives good results (Manning, 2008). The probability that a given document is to be judge relevant is given by:

$$P(R|x,q) = \frac{P(x|R,q).P(R|q)}{P(x|q)} \tag{3}$$

where $P(R = 1|x,q)$ and $P(R = 0|x,q)$ are the probabilities of retrieving a relevant or non-relevant document respectively.

2.2.3 Language Models

In the case of Language Models, a document is a good match for a query if a probabilistic generative model for the documents is capable of generating the query, which happens when the document contains the terms of the query more often.

Compared to the previously introduced discriminative probabilistic models, instead of modeling the probability of relevance of a document d for some query q, i.e. $P(R = 1|q_{k1}, d_m)$, the Language Model builds a model θ_d from each document d and thereafter ranks the documents based on the probability of the document model having generated the query, i.e. $P\left(q|\theta_d\right)$.

A document model can generate a query through various methods, including finite automata, n-grams or even by using probabilistic estimates based on the Bayes theorem (Manning, 2008), where the probability of a document is interpreted as the likelihood of having query q being produced from the document d. This probability is determined by inferring a document model θ_d for each document and then computing the probability of the query given the document model, $P(q|\theta_d)$ (Balog, 2008).

In order to construct a document model, a document d is represented as the probability distribution of the vocabulary terms over the document, i.e. $P(t|d)$. The *maximum likelihood* estimate of a term, which is given by its relative frequency in the document, provides the simplest method for inferring an empirical document model.

Considering a common approach, where the document model θ_d is a *unigram* language model, the probability of a query q is the product of the individual term probabilities, and is given by Equation 4.

$$P(q|\theta_d) = \prod_{t \in q} p(t|\theta_d)^{n(t,q)} \tag{4}$$

In the above equation, $n(t,q)$ corresponds to the number of times term t occurs in the query q. However, this approach has the limitation of giving a zero probability when one or more query terms do not appear in the document and therefore smoothing techniques need to be applied (Balog, 2008).

A common way to apply smoothing techniques to language models is to use a smoothing parameter λ which assigns some weight to a document in the collection, even if a given query term does not appear in the document. This parameter assumes values between 0 and 1.

A known approach to build a document model using smoothing techniques is given by the Jelinek-Mercer smoothing method, which is given by Equation 5.

$$P(q|\theta_d) = \prod_{t \in q} (1 - \lambda_t) P(t|d) + \lambda_t P(t) \tag{5}$$

The book by Manning (2008) provides additional information on the subject of language models for IR.

2.3 Link-Based Information Retrieval

The first generations of search engines used textual contents exclusively to generate the ranking lists. However, applying these methods to the web did not provide good results. When considering the web, too many documents may contain the query terms and it is hard to discriminate which among them are in fact the most relevant. Modern web search engines use linkage data together with the document's textual contents. They are able to gather information mined from the documents themselves, as well as information from the linkage structure of the web (Sebastiani, 2003). This linkage structure is composed of hypertext links from source documents to target documents, and it can be seen as a citation network.

More formally, a citation network is a directed graph composed of links between source nodes (citing) to target nodes (cited). This can be useful to determine the relevance of a document, because each time a document (e.g. an academic article or a web page) is cited, it reflects its quality/significance/impact to the author of the document making the citation. The links between documents are called *inLinks* and are hypertext links pointing to a web page. The area of link-based information retrieval borrowed techniques from bibliometrics (i.e., the study of citation patterns in academic publications) and developed approaches for estimating document importance and similarity between documents, using linkage information.

One of the best known linkage analysis techniques currently used in information retrieval is PageRank, an algorithm proposed for Google which assumes that the prestige of a node is proportional to the sum of the PageRank scores from the nodes that link to it. Thus, a node (i.e., a document) that is connected to nodes with a high PageRank value also receives a high ranking.

Equation 6 shows how to compute the PageRank (P_r) score recursively for a node i:

$$P_r(i) = \frac{0.5}{N} + 0.5 \sum_{j \in inLinks(L,i)} \frac{\alpha_j P_r\left(j\right)}{outLinks\left(L, j\right)} \tag{6}$$

In the formula, N is the number of nodes in the graph, $\alpha = 1, 2, ...L$ is the weight of each node and L represents the direct links between a source node and a target node. There are many extensions of the PageRank algorithm which will be addressed in detail over Section 5.3.

To measure the similarity of two nodes in a graph (e.g. two documents or two authors), it is often useful to take into account the citation patterns between the elements of the collection. Two popular similarity metrics based on citation networks are bibliographic coupling and co-citation.

In Bibliographic Coupling, two documents are similar if they share some references and the strength of that similarity is measured by the number of common references (Kessler, 1963). Figure 1 shows a directed graph illustrating the concept of bibliographic coupling. In this figure, documents A and B have a bibliographic coupling of 3.

Co-Citation is a variation of Bibliographic-Coupling originally proposed by Small (1973). Two documents d_1 and d_2 are similar if there are other documents citing both d_1 and d_2 and the strength of their similarity depends on the frequency with which the two documents are cited together in a same document. Figure 1 shows a citation graph illustrating the co-citation approach. In that figure, documents A and B have a co-citation score of 3.

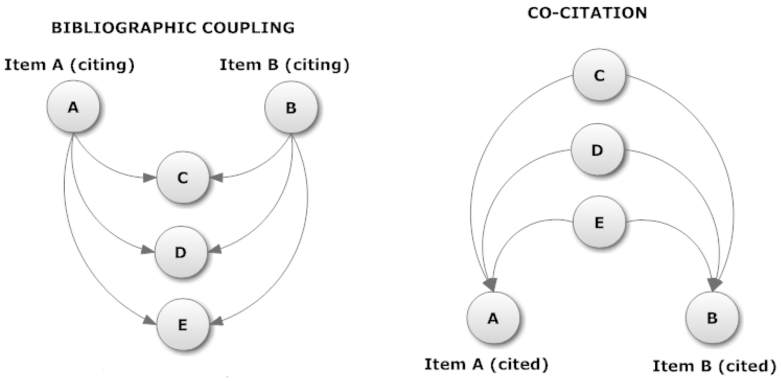

Fig. 1. Bibliographic Coupling and Co-Citation (Garfield, 2001)

2.4 Evaluation in Information Retrieval

The evaluation of IR systems is typically made through laboratory experiments relying on a document collection. In the field, this is known as the Cranfield methodology, since the first experiments that used this method were made at the Cranfield Institute of Technology in the late 1950. According to Rowe et al. (2010), the components of this methodology are:

- A collection of documents to be searched (i.e., the test collection);
- A series of queries answered by some documents in this collection, representing possible information needs of the users;
- The results of the IR system when trying to match the information needs with the document collection (i.e., the retrieved documents);
- Performance measures based on relevance judgments, stating which documents in the collection are relevant for each query.

Early experiments in the field were based on a complete relevance assessment of the document collection. Later experiments on the context of the Text Retrieval Conference (TREC) used much larger document collections, using a *pooling* methodology instead of complete relevant assessments. The TREC methodology is very robust and avoids the exhaustive assessment of all documents by the users (Belew, 2000). In the pooling method, the results submitted by the participants are used to form a pool of documents for each topic, by collecting the highly ranked documents from all submissions (Gozalo et al., 2000). This guarantees that the list of documents assessed for relevance is as comprehensive as possible (Sormunen, 2002). If a document is not included in the pool, then it remains unjudged and is assumed irrelevant (Belew, 2000). In order to enable evaluation, a binary scale is often used to determine the relevance of the documents. TREC states that one can only make binary judgments (i.e., determining whether a document is relevant) if there is one piece of supporting information in the document (no matter how small it is) that is relevant to the query (Sormunen, 2002).

The two most frequent and basic measures of effectiveness, based on binary relevance judgments, are Precision and Recall. These two measures are defined for the simple case where an IR system returns an unordered set of documents for a query.

Precision measures the fraction of the returned results which are relevant to the query and is given by:

$$Precision = \frac{|\,\{RelevantDocuments\} \bigcap \{RetrievedDocuments\}\,|}{|RetrievedDocuments|} \tag{7}$$

Recall measures the fraction of the relevant documents in the collection which were returned by the system and is given by:

$$Recall = \frac{|\,\{RelevantDocuments\} \bigcap \{RetrievedDocuments\}\,|}{|RelevantDocuments|} \tag{8}$$

In a ranked retrieval context, appropriate sets of retrieved documents should be given in the top k retrieved documents. In order to discriminate between ranked lists, it is necessary to extend these basic measurements or create new ones (Manning, 2008). Four other measures which have into account the top k retrieved documents are the precision at rank k, the mean average precision, the normalized discount cumulative gain and the mean reciprocal rank.

Precision at rank k is used when a user wishes only to look at the first k retrieved documents. The precision is calculated at that rank position through Equation 9.

$$P@k = \frac{r\,(k)}{k} \tag{9}$$

In the formula, $r(k)$ is the number of relevant documents retrieved in the top k.

The Mean Average Precision (MAP) directly considers the order in which relevant documents are returned by a system. For some query, the Average Precision is the average of precision values obtained in the top K documents returned by a system. The MAP is nothing more than the mean value of the Average Precision computed for various queries (Manning, 2008)

$$MAP = \frac{1}{|Q|} \sum_{j=1}^{|Q|} AP(j) \tag{10}$$

In the formula, AP is the Average Precision, which is given by:

$$AP = \frac{1}{k} \sum_{i=1}^{k} Precision(i) rel[i] \tag{11}$$

where $rel\,[i]$ is a function which assumes the value 1 if $rel\,[i] = TRUE$ and 0 otherwise.

Precision at rank k and Mean Average Precision are both based on binary judgments. The NDCG metric, on the other hand, is based in non binary judgments. It not only gives more importance to the top k retrieved documents, but also takes into account how these k documents are ordered. It measures the degree of relevance of documents based on their position in the result list produced by the system for some query. It assumes that highly relevant documents are more valuable than marginally relevant ones, and the greater the ranked position of a relevant document, the less valuable it is for the user, because the more likely it is that the user will ever examine the document (Järvelin and Kekäläinen, 2002).

In short, NDCG is used to emphasize the highly relevant documents which appear on top of the list of results returned by a query. It is given by:

$$NDCG_p = Z_p \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2 (1 + i)} \tag{12}$$

In the formula, rel_i is the relevance score assigned to document i and Z_p is the normalization factor calculated by normalizing the set of the top retrieved documents. For a perfect ranking, NDCG assumes the value 1 (Manning, 2008).

The Mean Reciprocal Rank (MRR) is a measure used to evaluate the correctness of one relevant document in a list of possible responses to a query. In Equation 13, for a query q, the rank position of its first relevant document is denoted as $rank_i$ and the Reciprocal Rank is given by the multiplicative inverse of the rank of the first correct answer, that is $\frac{1}{rank_i}$. It follows that the Mean Reciprocal Rank is the mean value of the Reciprocal Ranks of the results for a set of queries (Sanderson, 2010).

$$MRR = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{rank_i} \tag{13}$$

For further information about these evaluation metrics, the reader is referred to (Manning, 2008).

3 Heuristic Approaches for Expert Search

Although expert search is a recent concern in the information retrieval community, there are already many research efforts addressing this specific task. In this section, the most relevant works in this area are described. After a careful analysis of the related literature, I suggest to classify the different approaches to the expert search problem into five categories, namely (i) candidate-based, (ii) document-based, (iii) hybrid, (iv) graph-based and (v) learning to rank approaches. Learning to rank approaches eventually combine features derived from models based on the other types. They will be described in Section 4. The remaining classes of expert search approaches will be detailed in the following subsections.

3.1 Candidate-Based Approaches for Expert Finding

In these approaches, the system gathers all textual information about a candidate and merges it into a single document (i.e., the profile document). The associations between each document in the collection and a candidate are uncovered through the occurrence of personal identifiers of the candidate experts, such as names or email addresses, in the original documents. The profile document can be seen as the representation of the candidate's knowledge and is ranked by the probability of the candidate given the query topics. This approach assumes that the document and the candidate are conditionally independent and is based on the assumption that the more a candidate talks about some topic, the bigger are the chances of him being an expert in that topic.

One of the first expert finding systems following this approach was proposed by Craswell et al. (2001). The system was based on a standard web search engine and the ranking was determined by text similarity measures computed between the query and the expertise evidence from the profile document.

Another well known candidate-based model was formalized by Balog et al. (2006). Their approach offers a general probabilistic framework for modeling the expert finding task, which can be extended in order to provide stronger associations between candidates and topic terms. This approach is usually referred to as *Model 1* and it uses Language Models to rank candidates according to the probability of the respective candidate model, obtained from the profile document, generating the query topics. Figure 2 shows the general framework of this model.

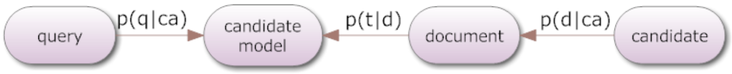

Fig. 2. Candidate-Based Approach (based on Balog (2008))

Petkova and Croft (2006) presented a general approach for representing the knowledge of a candidate expert as a mixture of language models from associated documents. Instead of merging all evidences of a candidate into one single profile document, they formed a group with the set of documents which are associated with a candidate and weighted the contribution of each document independently. The candidates were ranked with ad-hoc retrieval methods. Although one may think that this approach is very similar to the well known document-based approach, it has some differences that put it in the category of candidate based approaches. In document-base expert finding, the system first gathers all documents which contain the query topics and only then it extracts the candidate names from that set. What Petkova and Croft (2006) proposed was a system that first gathers all documents which are associated with some candidate. With that candidate set, they generated the ranked list of results according to the query topics.

Later, Petkova and Croft (2007) introduced the idea of dependency between experts and document terms. They formalized a candidate-centered document representation which uses positional information (i.e., the distance between a candidate name and the query terms) to weight the strength of the associations between terms and candidates. Specifically, they used proximity kernels to account with this distance, by fitting a multinomial density function around the occurrences of a candidate in a document.

Balog et al. (2009) also explored the positional information and formalized a candidate-based document representation which includes a window surrounding the candidate's name. The idea behind this method is that the closer a candidate is to a term, the more likely is that term to provide evidence of his knowledge.

In order to improve retrieval performance in adhoc retrieval tasks, query expansion methods (a.k.a. pseudo-relevance feedback) are often used. These techniques can also be applied in the context of expert finding. The main idea of query expansion consists in examining the top retrieved documents (i.e., the pseudo-relevance set) by an IR system, afterwards using specific information of these documents in order to re-weight the query terms and consequently improve the ranking of retrieved documents (Salton and Buckley, 1997).

Macdonald and Ounis (2007b) employed two *Divergence From Randomness* techniques to extract informative terms from the pseudo-relevance set. They also showed that query expansion techniques can be used in the context of a voting approach for the expert finding (Macdonald and Ounis, 2006). They proposed the candidate centric query expansion approach, where the pseudo-relevance set is taken from the final ranking of candidates generated by a query. One can see this model as the pseudo relevance feedback extension for the candidate-based approach. However, they showed that the candidate centric query expansion method performed poorly when compared to the document centric one.

Later, when investigating the poor results from the candidate centric approach, Macdonald and Ounis (2007a) discovered that it suffered from topic drift. Since a candidate can have several unrelated expertise areas, when performing pseudo-relevance feedback those expertise areas will gain relevance and will therefore lead to wrong results. The authors proposed to use a measure of cohesion to overcome that problem, obtaining better results.

Serdyukov et al. (2007) proposed to apply query expansion through the usage of pseudo-relevance feedback language models obtained from the top retrieved documents and the top retrieved candidates. This approach takes into account other expertise areas related to an expert without suffering from topic drift.

Fang and Zhai (2007) applied the *probabilistic ranking principle* to develop a general framework from which the candidate-based and document-based models for expert finding could be derived. They also showed how query expansion techniques, such as the association of different weights to each candidate representation and the topic expansion in order to give more information terms to the original query, can improve the performance of the models from the framework.

Balog and de Rijke (2008b) tried to extend the candidate-based model by adding non-local evidence, i.e., expertise evidence obtained using information that is not available in the immediate proximity of a candidate expert's name in a document. The idea is that the mere co-occurrence of a person with a query term is not necessarily an indication of expertise of that person on the topic. If a candidate is associated with a large number of documents, then the probability of that candidate associated to some particular document should be low, because it does not contribute significantly as evidence. In the same way, if a candidate is associated with many documents which are semantically similar, and then is associated with another document which is semantically different from the others, then the probability of that document being associated with the candidate should also be low.

3.2 Document-Based Approaches for Expert Finding

In these approaches, the system starts by gathering all documents which contain the expertise topic terms included in the query. Afterwards, the system uncovers which candidates are associated with each of those documents.

Contrary to the candidate-based approaches that were previously presented, which represented the candidate's knowledge with a single profile document, the document-based approach is based on ad-hoc information retrieval methods. First, the system gathers all documents which are relevant to the information need. Then, for each relevant document, the system determines the probabilities of the candidates being associated with it. Finally, the system adds up the individual relevance probabilities of each candidate and returns the list of possible candidate experts. The idea behind this approach is that if there is a candidate name in a piece of text where the query topics occur often, then it is probable that the candidate is an expert in that topic.

This approach was first used by Cao et al. (2006) in the Text REtrieval Conference of 2005. They proposed a two-stage language model where the first stage, called the relevance model, determines whether a document is relevant to the query topics or not. The second stage, called the co-occurrence model, determines whether or not a query topic is associated with a candidate. To improve this model, they added co-occurrence sub-models which give more weight to a document when a query and a person occur within the same window of text. For an extreme case, this window can have the size of the entire document. They also applied sub-models which give different weights according to the positions of the document that a candidate could appear.

A well known document-based model was formalized by Balog et al. (2006). This approach is commonly referred to as *Model 2* and language models are used to rank the candidates according to the probability of a document model having generated the query topics. Figure 3 shows the general framework of this document-based model.

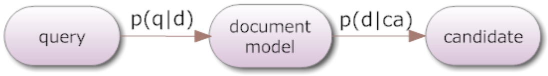

Fig. 3. Document-Based Approach (based on (Balog, 2008))

Balog et al. (2009) also explored the positional information and formalized a document representation which includes a window surrounding the candidate's name. The idea behind this approach is that the closer a candidate is to a term, the more likely that term is an evidence of his knowledge.

Balog et al. (2009) tested exhaustively the candidate-based and document-based approaches. They concluded that the document-based approach outperforms the other one. They also tested these approaches with positional information, and this time, the candidate-based approach achieved better results.

Query expansion techniques were also used in the context of the document-based approaches for the expert finding task. Macdonald and Ounis (2007b) proposed the document centric query expansion approach, where after ranking the documents relatively to a query, a pseudo-relevance set is taken from the top ranked documents of the document ranking. *Divergence From Randomness* techniques are then used in order to extract informative terms from the pseudo-relevance set composed of the top-retrieved documents.

Data fusion techniques have also been tested on the development of document-based expert search systems. For instance, Macdonald and Ounis (2008) used data fusion techniques to combine different document rankings into a single ranking and this approach enabled better results than the approaches based on language models.

Macdonald and Ounis (2006) were also the first authors to formalize a voting framework combined with data fusion techniques to be applied in the context of expert finding. In this framework, the system first ranks all documents respectively to the query topics. Then, each candidate associated with the top retrieved documents receives an implicit vote which states the expertise relevance of the candidate relatively to the query. The ranking of each candidate is given by the aggregation of the votes of each document. The authors tested their framework with 12 different voting techniques, where each of the voters represented sources of evidence that can be derived from the top retrieved documents. These voting techniques can be seen as approaches for aggregating scores.

Later, Macdonald et al. (2008) applied clustering techniques to the candidates profiles to identify the main interests of each candidate, under the assumption that the main expertise areas will be given by the largest clusters. Only the profiles of candidates who had a number of associated documents bigger than a given threshold θ were clustered and they used the Cosine between the average of each cluster as the clustering distance. The clusters were ranked by the number of documents that they contained and the top K clusters were chosen as representatives of the candidate's expertise. The authors combined these clusters with their voting approach, assuming that votes for candidates retrieved by documents belonging to the largest clusters were given more weight and, consequently, the retrieval performance of the system would improve. They also included techniques based on URL length and inLinks to identify the candidate homepages, because they contain personalized and professional information about the candidate.

3.3 Hybrid Approaches for Expert Finding

In the case of hybrid approaches, the system combines the ideas of the candidate and document-based approaches. This means that the system starts by ranking the candidate experts through their own language models, which were generated from the top retrieved documents. Then, the level of expertise on the search topic is given by the frequency of the candidate's mentions in the top ranked documents.

The most important model which falls into this category, as far as I know, was formalized by Pavel Serdyukov in his PhD thesis (Serdyukov, 2009).

He proposed the *Person-Centric model* which does not follow the independence assumption used in the candidate and document-based approaches. Instead, this model assumes that candidates are responsible for generating the terms within retrieved documents, which means that there should be a dependency relation between the query terms and a candidate.

The idea behind this model is that if a candidate is mentioned in a document, then it might be responsible for its content, either explicitly as the author of the document or implicitly as a recipient. This leads to the fact that a candidate expert is a strong indicator for a document topic. Figure 4 shows the general framework of this model.

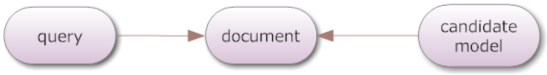

Fig. 4. Person-Centric Approach

The person-centric model starts just like a document-based approach, by retrieving the top ranked documents according to the query topics. Since these documents do not have a language model, the system requests to the candidate experts to generate the document terms using their own language models. Then, it ranks candidates just like in the candidate-based approach, by combining (i) the probability of generation of the query by the candidate's language model, and (ii) the prior probability of being an expert expressed in terms of a candidate's activity in the top retrieved documents. To build the candidate language models, two methods based on the *Expectation-Maximization* algorithm were proposed, namely one that considered that the probability of the association between a candidate and a document was fixed and fully dependent on the field of a document where the candidate appeared, and another that obtained those probabilities dynamically by predicting the contribution of specific persons to a document with basis on their intermediately estimated language models (Serdyukov, 2009).

3.4 Graph-Based Approaches for Expert Finding

The approaches presented so far do not take into account the linkage structure between candidates and documents. They are only focused in uncovering associations between topics and persons by analyzing the documents' textual content which a candidate is related to. Hence, they miss information by not considering the expertize of directly and indirectly linked candidates, and the relevance of documents which are in a near neighborhood of a candidate.

Using graph-based approaches, the system starts by ranking documents according to the query topics. From this set of documents, the system then extracts the contained candidates into another set. In this model, the documents and the candidates can be seen as nodes and their containment relations can be seen as directed edges. Together, they form an *expertise graph*.

Campbell et al. (2003) developed an expert search task approach for email documents, arguing that these documents are a valuable source of expertise. The approach first collected all emails which contained the query topics. Then, this set of retrieved documents enabled the creation of a directed social graph from email corpora, using people as nodes and the email headers *from/to* as edges. Finally, they applied a modified version of the HITS algorithm (Kleinberg, 1999) that only considered authority scores to rank the candidates according to their expertise level.

Zhang et al. (2007) applied expert search approaches in a web forum. Since each thread in a forum discusses a specific topic, they formalized these threads as *post/reply* directed social graphs, where each participant was a node and the edges corresponded to directed links associating a user posting a question to a user replying to it. They also applied network-ranking algorithms like PageRank and HITS, concluding that the characteristics of the network structure matter. For instance, in the forum on which they worked on, persons replying to many users are not necessarily experts, because they could be answering only to *newbies*. They compared their results against human judgments and they concluded that using PageRank and HITS provided better results than using a standard document-based approach.

Chen et al. (2006) made a similar study, where they also built social networks based on *post/reply* headers from forum corpora. They ranked the nodes in the network with PageRank and HITS. Instead of using an online community forum like Zhang et al. (2007), they used the W3C corpus of the TREC 2006 enterprise track. They concluded that for this dataset, both PageRank and HITS did not performed very well when compared with a standard document-based approach.

Instead of analyzing community based forums or email corpora, Liu et al. (2005) focused on finding authoritative scientists in digital libraries. They modeled a co-authorship network where each node is a person and each edge links a person to another if they have worked together. They applied several of the measures that are traditionally applied in undirected graphs, namely clustering, degree, closeness, betweenness, PageRank, and others. According to the authors, the degree measures how many connections tie authors to their immediate neighbors in the network. Closeness focuses on how close an author is to all other authors. And betweenness is based on determining how often a particular node is found on the shortest path between any pair of nodes of the network. The authors also defined a new measure based on PageRank, the *Author Rank*, to estimate the impact of an individual author in the network. The conclusion that they have taken is that these measures perform well in the identification of important authors. However, these co-authorship networks do not take into account the level of expertize of the authors, because they ignore the relations which are only in the context of a certain topic.

Serdyukov et al. (2008) represented candidate experts and documents in *expertise graphs* and modeled the principle of expert finding by three types of probabilistic random walks: finite, infinite and absorbing. To model the *expertise graph* they first applied a document-based approach where the system retrieves the most relevant documents which contain the query topics. Then, from this set of relevant documents, the contained candidates are extracted. Candidates and documents become nodes in the *expertise graph* and their containment relations become directed edges. Then, they exploited all known connections in the graph by including further links, such as expert-to-expert, making the graph very dense and enabling a better propagation of relevance, consequently enabling persons to receive expertise evidence from documents, even if these documents are not in an immediate proximity. Finally, they experimented separately the three random walk methods and concluded that *expertise graphs* with finite random walks provide much better results than infinite or absorbing random walks.

3.5 Comparison Between the Previously Presented Approaches

As already mentioned, TREC provided a platform for researchers to test their expert finding approaches. Table 1 shows the top three best-performing approaches in the 2005-2008 editions of the TREC Enterprise Track. Analyzing Table 1, one can see that the TREC-2005 edition was the one which had the

TREC Top 3 Approaches	MAP	MRR
(2005) 1st Fu et al. (2006)	0.2749	0.7268
(2005) 2nd Cao et al. (2006)	0.2688	0.6244
(2005) 3rd Yao et al. (2006)	0.2174	0.6068
(2006) 1st Zhu et al. (2007)	0.6431	0.9609
(2006) 2nd Bao et al. (2007)	0.5947	0.9358
(2006) 3rd You et al. (2007)	0.5639	0.9043
(2007) 1st Fu et al. (2008)	0.4632	0.6333
(2007) 2nd Duan et al. (2008)	0.4427	0.6131
(2007) 3rd Zhu et al. (2008)	0.4337	0.5802
(2008) 1st Balog and de Rijke (2008a)	0.4490	0.8721
(2008) 2nd Shen et al. (2008)	0.4214	0.7241
(2008) 3rd He et al. (2008)	0.4126	0.7611

Table 1. Retrieval performance of the top 3 best systems in TREC

lowest results. This is understandable, because it was the first time that there was an expert search competition. Most of the 2005 competitors used ad-hoc information retrieval methods to find ways of estimating the candidate's expertise from the dataset. As explained in the beginning of this work, expert finding is more difficult than an ad-hoc document retrieval task and traditional information retrieval methods are not enough.

The 2006 edition of TREC had the best results of all editions. Expert finding started to receive a lot of attention from researchers and many different approaches based on language models were developed. Some of the main approaches for expert finding were first shown in this edition, such as the candidate-based and the document-based approaches. These new formalizations were found to be very good when compared to the basic ad-hoc information retrieval methods presented in TREC-2005.

In the editions of 2007 and 2008, the results decreased when compared to the 2006 edition. In 2007, TREC changed the dataset to the publicly available pages of an Australian organization called CSIRO. TREC-2007/2008 participants were provided with only a structural template of email addresses used by CSIRO employees. Most participants had to get around spam protection, check if similarly looking addresses belonged to the same employee and filter non-personal addresses (Serdyukov, 2009). This dataset may have turned the expert finding task a little more difficult and, for that reason, the results were not as good as in the 2006 edition.

Some of the approaches mentioned in the previous sections, although not having achieved one of the top three places in TREC, are very well formalized and are easy to understand. Besides, many of the models from the winning teams are more effective extensions of the base models described in the previous sections. A comparative analysis of the previous systems is also shown in Table 2.

TREC Edition	2005	2005	2006	2006	2007	2007
Approaches Referred in Section 3	MAP	MRR	MAP	MRR	MAP	MRR
Petkova and Croft (2006)	0.2850	0.6496	-	-	-	-
Balog et al. (2006)	0.1894	0.2434	-	-	-	-
Chen et al. (2006)	-	-	0.4949	-	-	-
Petkova and Croft (2007)	-	-	0.6193	0.9541	-	-
Fang and Zhai (2007)	0.204	-	0.465	-	-	-
Macdonald and Ounis (2007b)	0.2231	-	0.5689	-	-	-
Macdonald and Ounis (2007a)	0.2271	-	0.5783	-	-	-
Macdonald and Ounis (2008)	0.2917	-	0.6571	-	-	-
Macdonald et al. (2008)	0.2324	-	10.5657	-	0.4319	0.5742
Balog and de Rijke (2008b)	-	-	-	-	0.5465	0.2880
Serdyukov et al. (2008)	-	-	0.413	0.807	0.407	0.566
Balog et al. (2009)	0.2725	0.6800	0.4697	0.9558	0.4633	0.6236
Serdyukov (2009)	0.1235	0.4700	0.4125	0.8120	-	-

Table 2. Retrieval performance of the approaches described in Section 3

4 Learning to Rank for Expert Finding

Information retrieval essentially deals with ranking problems. A system checks the documents of a collection and ranks them according to their degree of relevance to a user query expressing the information need. The major difficulty faced by an IR system resides in the computation of these relevance estimates. In the previous sections, it was presented a set of approaches which required the manual tuning of parameters (e.g., the λ in the language models, the k_1 and b in BM25 and the α in PageRank). In order to get a good ranking performance, these parameters need to be tuned for the validation set. And that is not a trivial task. For that reason, there was an increasing growth of interest in automatic ranking systems.

Learning to Rank for Information Retrieval is a particular approach to these ranking problems which automatically tunes parameters and builds ranking models, through the usage of hand labeled training data (e.g., document collections containing relevance judgments for specific sets of queries, or information regarding user-clicks aggregated over query logs), this way leveraging on data to combine different estimators of relevance in an optimal way. The learned raking model can then sort documents according to their degree of relevance to a given query. In the scope of expert finding, learning to rank (L2R) methods can automatically help to sort candidate experts according to the query topics.

Figure 5 provides an illustration of the general approach which is commonly used in most supervised learning to rank methods. It consists in two separate steps, namely training and testing.

In this framework, $q_i \, (i = 1, ..., n)$ corresponds to the set of n queries for the training step, $x^{(i)} = \left\{ x_j^{(i)} \right\}_{j=1}^{m(i)}$, with $m\,(i)$ being the number of documents associated with query q_i, are the feature vectors associated to each query and

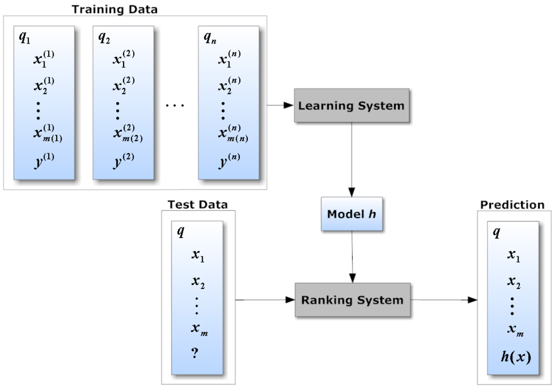

Fig. 5. The general Learning to Rank framework (Liu, 2009)

y^i $(i = 1, ..., n)$ is the corresponding relevance judgment. When applying a specific learning method to this training set, the system combines the different estimators of relevance in an optimal way, learning the corresponding ranking model. During the learning process, a loss function is applied to measure the inconsistency between the hypothesis space h and the ground truth label y.

In the test step, the learned ranking model is applied to a new given query in order to sort documents according to their relevance to the information need. The ranked document set is then returned as a response to the query.

Liu (2009) classified the existing learning to rank algorithms into three major approaches, namely the *pointwise*, *pairwise* and *listwise* approaches.

4.1 The Pointwise L2R Approach

In pointwise approaches, since the relevance degree can be regarded as a numerical or ordinal score, the learning to rank problem is seen as a regression, classification or ordinal regression problem. The idea behind the *pointwise* approach is that given feature vectors of each single document of the training data for the input space, the relevance degree of each of those documents is predicted with scoring functions which can sort all documents and produce the final ranked list. A loss function is applied in order to measure the individual inconsistencies between the scoring function and the ground truth labels.

Many supervised machine learning algorithms for regression, classification or ordinal regression can be readily used for this purpose.

In regression based algorithms, the relevance degree is seen as a real number. The two most representative algorithms which are based in regression are the Polynomial Regression Function (Fuhr, 1989) and the Subset Ranking with Regression (Cossock and Zhang, 2006).

In classification based algorithms, the ground truth label is not regarded as a quantitative value. Representative approaches are the Discriminative Model for

IR (Nallapati, 2004) and the Multi-Class Classification for Ranking (McRank), an algorithm proposed by Li et al. (2008).

The McRank algorithm uses a boosted ensemble of regression trees to learn a non-linear itemwise model for $P(z|q_i, e_j)$, i.e., the probability of a query-document pair (q_i, d_j) falling into each relevance bucket z. The interesting twist is that, instead of assigning relevance $\arg\max_z Pr(z|q_i, d_j)$, McRank assigns a score $\sum_z zP(z|q_i, d_j)$ to the jth document responding to query q_i, and then sorts the documents by decreasing score.

In ordinal based algorithms, the learning model considers the ordinal relationship of the ground truth labels. The aim of the ordinal regression problem is the use of thresholds in order to define a scoring function, discriminating the outputs of the scoring function into different ordered categories (Liu, 2009). The most representative algorithms of this approach are the Perceptron Based Ranking (PRanking), algorithm of Crammer and Singer (2001), and the Ranking with Large Margin Principles, algorithm of Shashua and Levin (2002).

The advantage of the *pointwise* approach is that it directly supports the application of existing theories and algorithms on regression or classification. However, the information order between documents cannot be considered in the training step, because the algorithm receives single documents as input. Since evaluation measures in IR are based in two fundamental criteria, namely query-level and position-based data, this loss of document order information is a major problem, because the evaluation measures which take into account the position of the documents will not be directly optimized and consequently the *pointwise* approach may unconsciously overemphasize unimportant documents.

4.2 The Pairwise L2R Approach

In pairwise approaches, since the relevance degree can be regarded as a binary value which tells which document ordering is better for a given pair of documents, learning to rank methods are seen as a classification problem.

The idea behind the *pairwise* approach is that given feature vectors of pairs of documents of the training data for the input space, the relevance degree of each of those documents can be predicted with scoring functions which try to minimize the average number of misclassified document pairs. The loss function used in the *pairwise* approach takes into consideration the relative order of each pair of documents and therefore it is very difficult to derive the order of the documents in the final ranked list.

Notice that the classification problem that this approach addresses is not the same as the one addressed by the *pointwise* approach. In the *pointwise* approach, the learning method operates on every single document. On the other hand, in the *pairwise approach*, the learning method operates on every two documents under investigation.

Several different *pairwise* methods have been proposed in the literature. The most representative methods are RankNet (Burges et al., 2005), RankBoost (Freund et al., 2003) and RankingSVM (Joachims, 2002).

The RankingSVM method builds a ranking model in the form of a linear scoring function, i.e. $f(x) = w^T x$, through the formalism of Support Vector Machines (SVMs). The idea is to minimize the following objective function over a set of n training queries $q_i{}_{i=1}^n$, their associated pairs of documents $(x_u^{(i)}, x_v^{(i)})$, and the corresponding relevance judgment $y_{u,v}^{(i)}$ over each pair of documents (i.e., pairwise preferences resulting from a conversion from the ordered relevance judgments over the query-document pairs).

$$\begin{aligned} &\min \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\sum_{u,v:y_{u,v}^{(i)}} \xi_{u,v}^{(i)} \\ \text{s.t. } & w^T(x_u^{(i)} - x_v^{(i)}) >= 1 - \xi_{u,v}^{(i)} \text{ , if } \ y_{u,v}^{(i)} = 1, \\ & \xi_{u,v}^{(i)} >= 0 \text{ , } \ i = 1, \dots, n \end{aligned} \tag{14}$$

Differently from standard SVMs, the loss function in Ranking SVM is a hinge loss defined over document pairs. The margin term $\frac{1}{2}\|w\|^2$ controls the complexity of the pairwise ranking model w. To allow some flexibility in separating the categories, SVM models have a cost parameter, C, which controls the trade off between allowing training errors and forcing rigid margins. It creates a soft margin that allows some misclassifications. Increasing the value of C increases the cost of misclassifying points and forces the creation of a more accurate model that may not generalize well.

The advantage of the *pairwise* approach is that it directly supports the application of existing theories and algorithms on classification. However, the loss function only takes into account the relative order of the pair of documents and therefore it is very difficult to derive the order of the documents in the final ranked list.

4.3 The Listwise L2R Approach

In the listwise approaches, the learning to rank problem takes into account an entire set of documents associated with a query as instances and trains a ranking function through the minimization of a listwise loss function defined on the predicted list and the ground truth list.

The idea behind the *listwise* approach is that, given feature vectors of a list of documents of the training data for the input space, the relevance degree of each of those documents can be predicted with scoring functions which try to directly optimize the value of a particular information retrieval evaluation metric, averaged over all queries in the training data. The loss function used in the *listwise* approach takes into consideration the positions of the documents in the ranked list of all the documents associated with the same query (Liu, 2009). This is a difficult optimization problem, because most evaluation measures are not continuous functions with respect to the ranking model's parameters. Continuous approximations or bounds on evaluation measures have to be used.

Several different *listwise* methods have been proposed in the literature. The most representative ones are SoftRank (Taylor et al., 2008), SVMmap (Yue et al., 2007), AdaRank (Xu and Li, 2007), ListNet (Cao et al., 2007), and ListMLE (Xia et al., 2008).

The SVMmap algorithm builds a ranking model through the formalism of structured Support Vector Machines (Tsochantaridis et al., 2005), attempting to optimize the metric Average Precision (AP). Suppose $x = \{x_j\}_{j=1}^{m}$ is the set of all the documents associated with a training query q, and $y_{u,v}^{(i)}$ represents the corresponding ground truth level. Any incorrect label of x is represented as y^c. The SVMmap optimization problem can be formalized as follows, where AP is used in the constraints of the structured SVM optimization problem.

$$\begin{aligned} &\min \frac{1}{2}||w||^2 + \frac{C}{n}\sum_{i=1}^{n}\xi^{(i)} \\ \text{s.t. } \forall y^{c(i)} \neq y^{(i)}, w^T\Psi(y^{(i)}, x^{(i)}) >= &\, w^T\Psi(y^{c(i)}, x^{(i)}) + 1 - AP(y^{c(i)}) - \xi^{(i)} \end{aligned} \tag{15}$$

In the constraints, Ψ is called the joint feature map, whose definition is:

$$\Psi(y, x) = \sum_{u,v:y_u=1,y_v=0} (x_u - x_v) \tag{16}$$

$$\Psi(y^c, x) = \sum_{u,v:y_u=1,y_v=0} (x_u^c, y_v^c)(x_u - x_v) \tag{17}$$

Since there are an exponential number of incorrect labels for the documents, it is a big challenge to directly solve the optimization problem involving an exponential number of constraints for each query. The formalism of structured SVMs efficiently tackles this issue by maintaining a working set with the constraints with the largest violation:

$$\mathrm{V}iolation \triangleq 1 - AP(y^c) + w^T\Psi(y^c, x) \tag{18}$$

Since in the SVMmap the computation of the most violated constraint is not very efficient, a new method was proposed in order to increase the algorithm's efficiency. The method proposed was the SVMndcg and argued that when the relevance at each position is fixed, the value of AP will be the same no matter which document appears at that position.

The SVMndgc algorithm also builds a ranking model through the formalism of structured Support Vector Machines (Tsochantaridis et al., 2005), attempting to optimize the metric of Normalized Discounted Cumulative Gain. The idea is similar to that of SVMmap, but it uses a different feature map and a different strategy for searching the most violated constraints.

The major advantage that the *listwise* approach has over the *pointwise* and the *pairwise* approaches is that its loss function takes into account the positions of the documents in a ranked list of all documents associated with the same query. Since evaluation measures in IR consider the position of the documents, this approach generally improves the performance of the learning methods.

4.4 Learning to Rank for Expert Finding Approaches

Many works on expert finding have been conducted. However, most of the methods proposed are based on language models and linkage information and very few studies have attempted to use learning to rank approaches.

The most related approach to the system that I propose to implement in the context of my MSc thesis is given by Yang et al. (2009), who proposed an expert search tool which employs learning to rank techniques to learn a function to rank candidate experts. Their framework consisted in three major modules, namely the *data preparation*, the *expert finding* and the *bólè search* modules. In the data preparation module, they extracted academic data from structured datasets, such as DBLP, and from unstructured web pages (e.g., the researchers homepages). The extracted and integrated information was stored in an academic network database. In the expert finding module, the authors used Ranking SVM as the supervised learning algorithm to rank candidates. Finally, they extended the expert finding system in order to perform a bólè search, that is, trying to identify the best supervisors in a specific filed. In the bólè search module, the authors tried to use the advantages of the generic labeled training data for expert finding. They proposed an approach which discovers the latent space while learning the ranking function. They defined features which were based in language models and in the expertise scores of a researcher. They evaluated the system using human judgments and obtained better results than the generic expert finding approaches based on languages models.

Expert finding can be seen as an entity ranking problem. In entity ranking, the goal is usually to rank entities in response to a query supported with a short list of entity examples (Pehcevski et al., 2008). By restricting the entity types to people, the expert finding task becomes a specialization of the entity ranking problem. Since there are very few approaches for expert search using learning to rank techniques, I will also describe some works in the area of entity ranking which use these automatic learning methods.

Qin et al. (2008) formalized entity ranking, as an optimization problem which should take into consideration not only the features of the objects and the query, but also the relations between the objects. Their method enhanced the attribute-based ranking function with parameterized regulation models of the relational graph, and the authors used SVM techniques to solve the learning problem (i.e. the optimization task).

Agarwal et al. (2006) proposed a framework for ranking networked entities by learning the parameters of Markovian walks in graphs, satisfying pairwise preference constraints between nodes. In the framework, they presented two different learning problems that relate to estimate conditional and absolute transition probabilities on each edge of the graph. In the first one, they assumed that the user has one or more hidden preferred communities with large edge conductance. In order for the learning algorithm to discover these communities with large edge conductance, they introduced a constraint maximum entropy network flow formulation whose dual can be solved efficiently using a cutting plane approach and a quasi-Newton optimizer. In the second problem, they assumed that edges have types that determine their conductance and those weights must be estimated by the learning function. In order to learn the conductances, they proposed an approximate gradient descent approach which can estimate relatively few global weights with a much smaller number of examples in the training data.

5 Features for Estimating Expertize

In my work, the features which will be considered to estimate the expertise level of a candidate, given a query, will fall into three categories, namely textual features, profile features and network features. The textual features are the same as the ones used in ad-hoc IR systems (e.g., TF.IDF and BM25). The profile similarity features are based on telling how important an author is in the scientific community, by looking at the number of citations and publications of the author. Finally, the network features correspond to importance and relevance estimates computed from the author's co-authorship and co-citation graphs.

5.1 Features Based on Textual Similarity

These features are based in the same assumption used in the document-based approaches for expert finding. If there is an author name in a piece of text where the query topics occur often, then it is probable that the author is an expert in that topic. In the scope of academic digital libraries such as DBLP, the relationships between documents and experts can be easily obtained from the authorship information. For each topic-expert pair, the set of textual features to be considered is as follows:

The Query Term Frequency (TF), corresponds to the number of times that each individual term in the query occurs in all the documents associated with the author. Equation 19 describes the TF formula, where $i \in Terms(q)$ represents the set of terms from query q, $j \in Docs(a)$ is the set of documents having a as author, $Freq(i, d_j)$ is the number of occurrences of term i in document d_j and $|d_j|$ represents the number of terms in document d_j.

$$TF_{q,a} = \sum_{j \in Docs(a)} \sum_{i \in Terms(q)} \frac{Freq(i, d_j)}{|d_j|} \tag{19}$$

The Inverse Document Frequency (IDF) is the sum of the values for the inverse document frequency of each query term and is given by Equation 20. In this formula, $|D|$ is the size of the document collection and $f_{i,D}$ corresponds to the number of documents in the collection where the i_{th} query term occurs.

$$IDF_q = \sum_{i \in Terms(q)} \log \frac{|D|}{f_{i,D}} \tag{20}$$

The Document Length (DL), corresponding to the total number of words in the documents associated with the author. The DL feature is query independent, so an author has always the same DL value no matter the query.

The Number of Unique Authors (NUA) in documents, corresponding to the total number of unique authors which are associated with documents containing at least one of the query terms. Since the NUA feature is author independent, all the authors for a given query have the same NUA value.

The Aggregated Okapi BM25, corresponding to the sum of the BM25 scores over all the documents associated with the author, for all the individual terms in the query. Equation 21 presents this BM25 formula, where $i \in Terms(q)$ is the set of terms from query q, $j \in Docs(a)$ corresponds to the documents having a as author, the number of occurrences of term i in document d_j is given by $Freq(i, d_j)$, $|d_j|$ corresponds to the number of terms in document d_j and $\mathcal{A}$ represents the average length of the documents in the collection.

$$\begin{aligned} BM25_{q,a} = \sum_{j \in Docs(a)} \sum_{i \in Terms(q)} \log \left(\frac{N - Freq(i) + 0.5}{Freq(i) + 0.5} \right) \times \\ \frac{(k_1 + 1) \times \frac{Freq(i,d_j)}{|d_j|}}{\frac{Freq(i,d_j)}{|d_j|} + k_1 \times (1 - b + b \times \frac{|d_j|}{\mathcal{A}})} \end{aligned} \tag{21}$$

The Average Okapi BM25 can be seen as the average of the Aggregated Okapi BM25 scores. In other words, it is the mean value of the BM25 scores computed over all the documents associated with the author, for all the individual terms in the query. The BM25 formula is given by Equation 21 and, in the computation of this feature, one simple replaces the summation over all documents $j \in Docs(a)$ by an average over these values.

The Maximum Okapi BM25, corresponding to the maximum value over the Aggregated Okapi BM25 scores. That is, it computes the maximum value of the BM25 scores computed all over the documents associated with the author, for all individual terms in the query. The BM25 formula is given by Equation 21 and, in the computation of this feature, one simple replaces the summation over all documents $j \in Docs(a)$ by the maximum of these values.

Years passed since the author's last publication containing the query terms. It is considered that authors who published more recently on the subject of the query are more likely to be considered experts than others who published long ago.

Years passed since the author's first publication containing the query terms. It is considered that the first authors publishing on the query subject are considered to be experts.

Time interval between the first and last of the author's publications which contain the query terms, in years. It is considered that authors with a longer publishing record on the subject of the query, are more likely to be considered experts.

In the computation of textual features, I will consider two different textual streams from the documents, namely (i) a stream consisting of the titles and the abstracts, and (ii) a stream using the entire textual contents of the articles.

5.2 Features Based on Profile Information

Since the task of expert finding is to search for the best people who are very knowledgeable about a certain topic, the features based on the profile information about the candidate expert play an important role. This profile informations is related to the amount of publications that an author has made and their impact in the scientific community. The idea is that highly prolific authors are more likely to be experts. The features based on the profile information are all query independent, that is, they have the same value for different queries. The set of profile features that I will be considered is as follows:

The Total Number of Publications of an author, considering that highly prolific authors should be more relevant.

The Total Number of Journal Publications of an author, considering that publications on journals are of highly quality and therefore very important to estimate relevance.

The Time Interval, in years, spanning through the dates of publications associated with the oldest and the newest articles of the author, under the assumption that authors with a longer career are more likely to be considered experts.

The Average Number of Publications per Year of an author, under the assumption that authors who maintain a regular publication activity should be considered more relevant. This is computed considering the interval of years from the oldest to the latest publication associated with the candidate expert.

The Average Number of Journal Publications per Year associated with an author, under the assumption that authors who maintain a regular journal publication activity should be more relevant.

5.3 Features Based on Connections between Experts

When considering academic digital libraries, linkage structures such as co-citation and co-authorship are usually used in order to discriminate documents and improve ranking scores. These graph-based structures can offer effective approaches for estimating the importance of the contributions of particular publications, publication venues or individual authors. The features that will be considered under this category are the following:

The Total Number of Citations for the Papers associated with an author which contain the query terms. It is considered that highly cited authors are more likely to be considered experts on the query topics.

The Average Number of Citations for the Papers associated with the author which contain the query terms. It is considered that authors who are cited very often are more likely to be experts in the subjects expressed in the query terms.

The Average Number of Citations, per year, over all papers belonging to an author that contain the query terms. It is considered that authors who are cited very often are more likely to be considered experts in the subject expressed in the query topics. The computation of this feature will only take into account the interval of years from the author's first publication to his last publication which contains a citation.

The Maximum Number of Citations for an Individual Paper associated with an author that contains the query terms. It is considered that authors of at least one highly influential paper are more likely to be experts.

The Total Number of Unique Collaborators who participated with the author in publications which contain the query terms. It is considered that authors who collaborate with many different people are more likely to be considered experts.

The Total Number of Citations to Papers containing the query terms which have an author from the same institution as the author, thus estimating the impact of the author's institution on the subject field of the query. It is considered that authors from high-impact institutions are more likely to be considered experts

The Hirsch Index of the author. This index measures both scientific productivity and his apparent scientific impact (Hirsch, 2005). An author has an Hirsch index of h if h of his N_p papers have at least h citations each, and the other $(N_p...h)$ papers have at most h citations each. It is considered that authors with a high Hirsch index are more likely to be considered experts.

The *H-b Index*, which is an extension of the Hirsch index for evaluating scientific topics in general (Banks, 2006). In the scope of expert finding, the scientific topic is given by the query terms. A query has an *h-b* index of i if i of the N_p papers containing the query terms have at least i citations each, and the other $(Np...i)$ papers have at most i citations each.

The Contemporary Hirsch Index of the author, is an extension of the Hirsch index that adds weights to each cited article with respect to its age. This means that older articles are given less weight than more recent ones (Sidiropoulos et al., 2007). A researcher has a Contemporary Hirsch Index h^c if h^c of his N_p articles get a score of $S^c(i) >= h^c$ each, and the rest $(N_p - h^c)$ articles get a score of $S^c(i) <= h^c$. For an article i, the score $S^c(i)$ is defined as:

$$S^c(i) = \gamma * (Year(now) - Year(i) + 1)^{-\delta} * |CitationsTo(i)| \tag{22}$$

According to Sidiropoulos et al. (2007), the parameters γ and δ should be set to 4 and 1 respectively, in order to get better results. These parameters mean that the citations for an article published during the current year count four times, the citations for an article published 4 years ago count only one time, the citations for an article published 6 years ago count 4/6 times, and so on.

The Trend Hirsch Index of the author is used to estimate the impact of a researcher's work in a particular time instance by assigning to each citation an exponentially decaying weight according to the age of the citation (Sidiropoulos et al., 2007). A researcher has a Trend Hirsch Index h^t if h^t of his N_p articles get a score of $S^t(i) >= h^t$ each, and the rest $(N_p - h^t)$ articles get a score of $S^t(i) <= h^t$. For an article i, the score $S^t(i)$ is defined as:

$$S^t(i) = \gamma * \sum_{\forall x \in C(i)} (Year(now) - Year(x) + 1)^{-\delta} \tag{23}$$

According to Sidiropoulos et al. (2007), the parameters γ and δ should be set to 1 and 4 respectively, in order to provide good results.

The Individual Hirsch Index of the author is used to reduce the effects of co-authorship of influential authors who contribute to the Hirsch Index of the author. This feature is computed by dividing the value of the standard Hirsch Index by the average number of authors in the articles that have a contribution to the Hirsch Index of the author (Batista et al., 2006).

The ***a*-Index** of the author, which measures the magnitude of his most influential articles. For instance, if an author has a Hirsch Index of h that has a total of $N_{c,tot}$ citations towards his papers, then he has an a-index of $a = N_{c,tot}/h^2$.

The g-Index of the author, also quantifying scientific productivity with basis on his publication record (Egghe, 2006). Given a set of articles associated with the author, ranked in decreasing order of the number of citations that they received, the g-index is the (unique) largest number such that the top g articles received on average at least g citations.

The Hirsch's Index of the institution of the author, under the assumption that authors from high impact institutions are more likely to be considered experts in the topics expressed in the query. An institution has a Hirsch Index of h if h of its N_p papers have at least h citations each, and the other $(N_p - h)$ papers have at most h citations each.

The a-index of the institution of the author, which estimates the impact of an institution by measuring the magnitude of its most influential articles. For instance, if an institution has a Hirsch Index of h and has a total of $N_{c,tot}$ citations towards its papers, then it has an a-index of $a = N_{c,tot}/h^2$.

The g-index of the institution of the author, also under the assumption that authors from high impact institutions are more likely to be considered experts. Given the set of articles associated with the institution, ranked in decreasing order of the number of citations that they received in the past, the g-index is the (unique) largest number such that the top g articles received on average at least a number of g citations.

Besides the above features, the ones that were defined in the previous work of Chen et al. (2007) will be also taken into consideration. In their work, Chen et al. (2007) considered a set of network features to estimate the influence of individual authors through the use of PageRank, which was already described in Section 2.3. The following set of features, based on PageRank, will be considered:

The Sum of PageRank Values associated to papers of the author which contain the query terms. Each citation link in the graph is given a score of $1/N$, where N is the number of authors in the paper. PageRank is afterwards computed on this weighted graph, representing citations between papers. The idea is that authors with a high PageRank score are more likely to be considered experts in the topics expressed in the query.

The Average of PageRank Values associated to papers of the author which contain the query terms. Each citation link in the graph is given a score of $1/N$, where N is the number of authors in the paper. The average of the PageRank is afterwards computed on this weighted graph, representing citations between papers.

The PageRank Value of an Author Computed Over a Directed Graph where each link represents the number of citations between authors. To each citation link between two authors a_1 and a_2, this approach gives a weight corresponding to the number of papers where author a_1 has cited author a_2.

The PageRank Value of an Author Computed over an Undirected Graph which represents collaborations between authors. In order to compute PageRank, the undirected graph will be first converted to a directed one by placing bi-directional links between each pair of nodes that are connected. To each citation link between two authors a_1 and a_2, this approach gives a weight corresponding to the number of papers where author a_1 collaborated with author a_2.

6 Thesis Proposal and Validation Plan

The main hypothesis behind my MSc proposal is that learning to rank approaches can be effectively used in the context of expert finding tasks, in order to combine different estimators of relevance in a principled way, this way improving over the current state-of-the art. Most of the works in the area of expert finding are based in the text content of the documents or in graph structures inferred from the community of experts. There are almost no works applying machine learning ranking models to rank experts and, as far as I know, no previous works in the field of expert finding have used academic indexes, such as the h and g indexes as estimators of relevance. The main research questions that I aim to answer are: (i) *Can learning to rank techniques be effectively applied in the field of expert finding?*, (ii) *Can profile features such as the h and g indexes, referred in Section 5.2, be effectively used as estimators of relevance to rank experts?* and (iii) *Can expert search techniques be effectively used in academic digital libraries related to areas such as Mathematics?*

To validate the aforementioned hypothesis and answer the above questions, I will build an expert search system prototype reusing existing implementations of state-of-the-art learning to rank algorithms, namely those implemented in the StructRank[1] package from Soumen Chakrabarti or in the extensions to the *weka* environment proposed by Lievens and De Baets (2010). I will make experiments with the application of representative learning to rank algorithms of different types, such as McRank, Ranking SVM, SVMmap and SVMndcg, in order to determine which is the best one in the task of expert finding within digital libraries of academic publications. I will also implement the methods responsible for computing the features listed in Section 5, reusing other Java software packages such as the LAW package for computing PageRank and its derivatives[2].

The validation of the prototype system requires a sufficiently large repository of textual contents describing the expertise of individuals within an organization or a specific area. This work will use datasets from three different domains, namely:

[1] `http://www.cse.iitb.ac.in/~soumen/doc/StructRank/`

[2] `http://law.dsi.unimi.it/index.php?option=com_content&task=view&id=35&Itemid=42`

- Publications from the computer science research community, specifically an enriched version of the DBLP dataset.[3]
- Publications from a national academic institution, specifically a dataset built from the institutional repository from Instituto Superior Técnico.[4]
- Publications from the mathematics research community, using resources made available in the context of the EuDML European project.[5]

DBLP data has been used in several previous experiments regarding citation analysis (Sidiropoulos and Manolopoulos, 2005, 2006) and expert search (Deng et al., 2008). It is a large dataset covering both journal and conference publications in the domain of computer science. The EuDML dataset, on the other hand, can enable the experimentation of the proposed techniques on a scientific field which has received less attention from the scientometrics and information retrieval communities, namely the field of Mathematics.

The prototype that will be implemented will consist in four modules, namely the feature computation module, the data management module, the learning module and the training module. The general framework of the prototype is given in Figure 6.

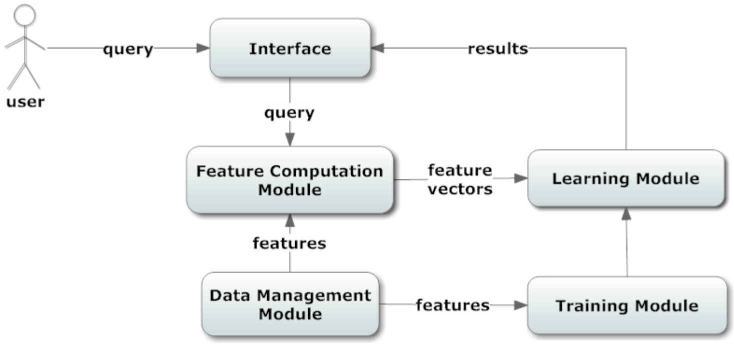

Fig. 6. General Architecture of the Proposed System

The data management module is essentially a database storing the information about the authors and their respective publication record. Query independent features, which were already presented in Section 5, will also be precomputed and stored in this database. The database scheme will follow the entity-relation diagram presented in Figure 7. In terms of implementation, I will use either MySQL or PostgreSQL since both these systems offer full-text search capabilities.

[3] http://www.informatik.uni-trier.de/~ley/db/
[4] http://www.ist.utl.pt/
[5] http://www.eudml.eu/

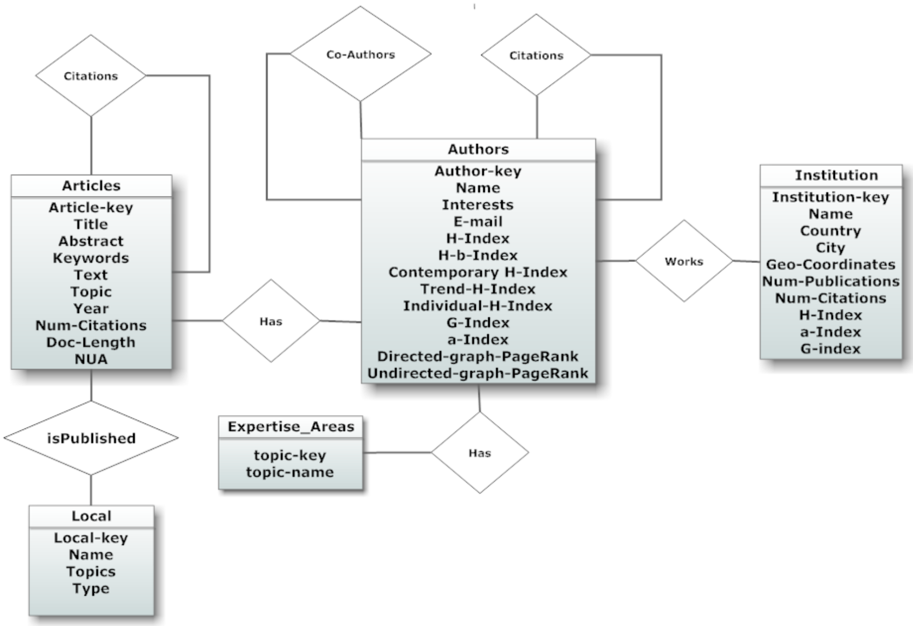

Fig. 7. Entity-Relation diagram of the database that will be used in the prototype

The database management module will use scripts, written in Java and XQuery, in order to extract the desired information from the papers. Since the information presented in the documents can be incomplete, the system will also attempt to extract profile information about each candidate from web sites such as Microsoft Academic Search. The more information there is about a candidate, the more accurate estimates can be made in order to determine his expertise.

The training module will use a dataset computed of <*topic, expert, relevance judgment*> triplets in order to learn a ranking model. The relevance judgments will be collected through different protocols in each of the envisioned application domains. For the EuDML application, experts involved in the EuDML project will be asked which are the top experts in various fields in the domain of Mathematics. On the other hand, for the DBLP application, I will use computer science awards as indicators of relevance, under the assumption that people who are in prize award lists are very likely to be considered experts in the topic of the award. To evaluate the system, I will use a set of these relevance judgments.

With basis in the above triplets, the system will compute all features associated with a candidate, this way building a set of feature vectors for each query. These features vectors will be used by the learning module in order to learn a ranking model just like described in Section 4.

At query time, users will interact with the system by submitting a query topic to the prototype's interface. When the topic is submitted, the feature computation module connects to the data management module in order to retrieve all candidates which have as expertise area the topic expressed in the query. Then, for each candidate from the retrieved set, a feature vector is built where query independent features are directly extracted from the database and query dependent features are computed in run time.

Finally, these feature vectors will serve as input for the ranking module, where the ranking model that was learned in the training module will be applied to each candidate vector, computing the ranking score. Candidates are then sorted in decreasing order of relevance and shown to the user.

7 Summary

The automatic search for knowledgeable people in the scope of specific communities or large organizations, with basis on documents describing people's activities, is an information retrieval problem that has been receiving increasing attention. These expert finding problems involve taking a short user query as input and returning a list of people, sorted by their level of expertise in what concerns the query topic. TREC provided a platform where researchers could test their techniques in the context of finding experts in an enterprise.

Several effective approaches to expert finding have been proposed in the literature, exploring different retrieval models and different sources of evidence for estimating expertise. However, the current state-of-the-art lacks in principled approaches for combining different sources of evidence in an optimal way. In the context of my MSc thesis, I will study the application of learning to rank approaches in the context of expert finding within academic digital libraries.

This document presented the fundamental concepts that will be used throughout my work and surveys on the information retrieval subjects of expert finding and learning to rank. The various expert retrieval models proposed in the literature were classified into five categories, namely (i) candidate-based, (ii) document-based, (iii) hybrid, (iv) graph-based and (v) learning to rank approaches.

The paper also introduced the area of learning to rank for information retrieval, presenting the pointwise, pairwise and listwise approaches. Moreover, the paper introduced a large set of features that can be used to estimate expertise in an academic setting, particularly useful when addressing the expert search problem through learning to rank. Finally, the paper detailed my thesis proposal and the envisioned validation plan.

Bibliography

Agarwal, A., Chakrabarti, S., and Aggarwal, S. (2006). Learning to rank networked entities. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.*

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval.* Addison Wesley.

Balog, K. (2008). *People Search in the Enterprise*. PhD thesis, University of Amsterdam, Faculty of Science, Mathematics and Computer Science.

Balog, K., Azzopardi, L., and de Rijke, M. (2006). Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*.

Balog, K., Azzopardi, L., and de Rijke, M. (2009). A language modeling framework for expert finding. *Information Processing and Management*, 45:1–19.

Balog, K., Bogers, T., Azzopardi, L., de Rijke, M., and van den Bosch, A. (2007). Broad expertise retrieval in sparse data environments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*.

Balog, K. and de Rijke, M. (2008a). Combining candidate and document models for expert search. In *Proceedings of the 2008 Text REtrieval Conference (TREC 2008)*.

Balog, K. and de Rijke, M. (2008b). Non-local evidence for expert finding. In *Proceeding of the 17th ACM conference on Information and knowledge management*.

Banks, M. G. (2006). An extension of the hirsch index: Indexing scientific topics and compounds. *Scientometrics*, 69:161–168.

Bao, S., Duan, H., Zhou, Q., Xiong, M., Cao, Y., and Yu, Y. (2007). Research on expert search at enterprise track of trec 2006. In *Proceedings of the 15th text retrieval conference (TREC 2005)*.

Batista, P. D., Campiteli, M. G., and Kinouchi, O. (2006). Is it possible to compare researchers with different scientific interests? *Scientometrics*, 68:179–189.

Belew, R. (2000). *Finding Out About*. Cambridge University Press.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*.

Campbell, C. S., Maglio, P. P., Cozzi, A., and Dom, B. (2003). Expertise identification using email communications. In *Proceedings of the 12th international conference on Information and knowledge management*.

Cao, Y., Liu, J., Bao, S., and Li, H. (2006). Research on expert search at enterprise track of trec 2005. In *Proceedings of the 14th text retrieval conference (TREC 2005)*.

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

Chen, H., Shen, H., Xiong, J., Tan, S., and Cheng, X. (2006). Social network structure behind the mailing lists: Ict-iiis at trec 2006 expert finding track. In *Proceedings of the 15th Text REtrieval Conference (TREC 2006)*.

Chen, P., Xie, H., Maslov, S., and Redner, S. (2007). Finding scientific gems with google's pagerank algorithm. *Journal of Informetrics*, 1(1):8–15.

Cossock, D. and Zhang, T. (2006). Subset ranking using regression. In *COLT*.

Crammer, K. and Singer, Y. (2001). Pranking with ranking. In *Advances in Neural Information Processing Systems*.

Craswell, N., de Vries, A., and Soboroff, I. (2005). Overview of the trec-2005 enterprise track. In *Proceedings of TREC-2005*.

Craswell, N., Hawking, D., Vercoustre, A.-M., and Wilkins, P. (2001). P@noptic expert: Searching for experts not just for documents. In *Ausweb*.

Deng, H., King, I., and Lyu, M. (2008). Formal models for expert finding on dblp bibliography data. In *Proceedings of the 8th IEEE International Conference on Data Mining*.

Duan, H., Zhou, Q., Lu, Z., Jin, O., Bao, S., Cao, Y., and Yu, Y. (2008). Research on enterprise track of trec 2007 at sjtu apex lab. In *Proceedings of the 16th text retrieval conference (TREC 2007)*.

Egghe, L. (2006). Theory and practise of the g-index. *Scientometrics*, 69(1):131–152.

Fang, H. and Zhai, C. (2007). Probabilistic models for expert finding. In *Proceedings of the 29th European conference on IR research*.

Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969.

Fu, Y., Xue, Y., Zhu, T., Liu, Y. Zhang, M., and Ma, S. (2008). The open university at trec 2007 enterprise track. In *Proceedings of the 16th text retrieval conference (TREC 2007)*.

Fu, Y., Yu, W., Li, Y., Liu, Y., Zhang, M., and Ma, S. (2006). Thuir at trec 2005: Enterprise track. In *Proceedings of the 14th text retrieval conference (TREC 2005)*.

Fuhr, N. (1989). Optimum polynomial retrieval functions based on the probability ranking principle. *ACM Transactions on Information Systems*, 7:183–204.

Fuhr, N. (1992). Probabilistic models in information retrieval. *Comput. J.*, 35:243–255.

Garfield, E. (2001). From bibliographic coupling to co-citation analysis via algorithmic historio-bibliography. Technical report, Drexel University, Philadelphia.

Gozalo, J., Verdejo, F., Penas, A., and Peters, C. (2000). User needs. Technical report, CLEF.

He, B., Macdonald, C., Ounis, I., Peng, J., and Santos, R. (2008). University of glasgow at trec 2008: Experiments in blog, enterprise, and relevance feedback tracks with terrier. In *Proceedings of the 2008 Text REtrieval Conference (TREC 2008)*.

Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. In *Proceedings of the National Academy of Sciences USA*.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20:422–446.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Kessler, M. (1963). Bibliographic coupling between scientific papers. *American Documentation*, 24:123–131.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of ACM*, 46:604–632.

Li, P., Burges, C., and Wu, Q. (2008). Learning to rank using classification and gradient boosting. In *Advances in Neural Information Processing Systems*.

Lievens, S. and De Baets, B. (2010). Supervised ranking in the weka environment. *Information Science*, 180:4763–4771.

Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations of Trends Information Retrieval*, 3:225–331.

Liu, X., Bollen, J., Nelson, M. L., and de Sompel, H. V. (2005). Co-authorship networks in the digital library research community. In *Information Processing and Management*.

Macdonald, C., Hannah, D., and Ounis, I. (2008). High quality expertise evidence for expert search. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval*.

Macdonald, C. and Ounis, I. (2006). Voting for candidates: adapting data fusion techniques for an expert search task. In *Proceedings of the 15th ACM international conference on Information and knowledge management*.

Macdonald, C. and Ounis, I. (2007a). Expertise drift and query expansion in expert search. In *Proceedings of the 16th ACM conference on Conference on information and knowledge management*.

Macdonald, C. and Ounis, I. (2007b). Using relevance feedback in expert search. In *Proceedings of the 29th European conference on IR research*.

Macdonald, C. and Ounis, I. (2008). Voting techniques for expert search. In *Knowledge Information Systems*.

Manning, C. D. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Mattox, D., Maybury, M. T., and Morey, D. (1999). Enterprise expert and knowledge discovery. In *Proceedings of the HCI 8th International Conference on Human-Computer Interaction: Communication, Cooperation, and Application Design*.

Maybury, M. T. (2006). Expert finding systems. Technical report, Center for Integrated Intelligent Systems, Bedford, Massachusetts.

Nallapati, R. (2004). Discriminative models for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*.

Pehcevski, J., Vercoustre, A.-M., and Thom, J. A. (2008). Exploiting locality of wikipedia links in entity ranking. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval*.

Petkova, D. and Croft, W. B. (2006). Hierarchical language models for expert finding in enterprise corpora. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*.

Petkova, D. and Croft, W. B. (2007). Proximity-based document representation for named entity retrieval. In *Proceedings of the 16th ACM conference on Conference on information and knowledge management*.

Qin, T., Liu, T.-Y., Zhang, X.-D., Wang, D.-S., Xiong, W.-Y., and Li, H. (2008). Learning to rank relational objects and its application to web search. In *Proceeding of the 17th international conference on World Wide Web*.

Ramos, J. (2001). Using tf-idf to determine word relevance in document queries. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.121.1424.

Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations of Trends Information Retrieval*, 3:333–389.

Rowe, B., Wood, D., Link, A., and Simoni, D. (2010). Economic impact assessment of nist text retrieval conference (trec) program. Technical report, National Institute of Standards and Technology.

Salton, G. and Buckley, C. (1997). *Improving retrieval performance by relevance feedback*, pages 355–364. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commununity of ACM*, 18:613–620.

Sanderson, M. (2010). Test collection based evaluation of information retrieval systems. In *Foundations and Trends of Information Retrieval*.

Sebastiani, F. (2003). *Advances in Information Retrieval - 25th European Conference on IR Research, ECIR*. Springer-Verlag.

Serdyukov, P. (2009). *Search for Expertise : Going Beyond Direct Evidence*. PhD thesis, University of Twente.

Serdyukov, P., Chernov, S., and Nejdl, W. (2007). Enhancing expert search through query modeling. In *Proceedings of the 29th European conference on IR research*.

Serdyukov, P., Rode, H., and Hiemstra, D. (2008). Modeling multi-step relevance propagation for expert finding. In *Proceeding of the 17th ACM conference on Information and knowledge management*.

Shashua, A. and Levin, A. (2002). Ranking with large margin principles: Two approaches. In *NIPS*.

Shen, H., Wang, L., Bi, W., Liu, Y., and Cheng, X. (2008). Research on enterprise track of trec 2008. In *Proceedings of the 2008 Text REtrieval Conference (TREC 2008)*.

Sidiropoulos, A., Katsaros, D., and Manolopoulos, Y. (2007). Generalized h-index for disclosing latent facts in citation networks. *Scientometrics*, 72(2):253–280.

Sidiropoulos, A. and Manolopoulos, Y. (2005). A citation-based system to assist prize awarding. *SIGMOD Rec.*, 34:54–60.

Sidiropoulos, A. and Manolopoulos, Y. (2006). Generalized comparison of graph-based ranking algorithms for publications and authors. In *Journal for Systems and Software*.

Small, H. (1973). Co-citation in the scientific literature: A new measurement of the relationship between two documents. *Journal of the American Society of Information Science*, 24:265–269.

Sormunen, E. (2002). Liberal relevance criteria of trec - counting on negligible documents. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*.

Taylor, M., Guiver, J., Robertson, S., and Minka, T. (2008). Softrank: optimizing non-smooth rank metrics. In *Proceedings of the international conference on Web search and web data mining*.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484.

Xia, F., Liu, T.-Y., Wang, J., Zhang, W., and Li, H. (2008). Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*.

Xu, J. and Li, H. (2007). Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*.

Yang, Z., Tang, J., Wang, B., Guo, J., Li, J., and Chen, S. (2009). Expert2bole: From expert finding to bole search. In *KDD'09*.

Yao, C., Peng, B., He, J., and Yang, Z. (2006). Cnds expert finding system for trec2005. In *Proceedings of the14th text retrieval conference proceedings (TREC 2005)*.

You, G., Lu, Y., Li, G., and Yin, Y. (2007). Ricoh research at trec 2006: Enterprise track. In *Proceedings of the15th text retrieval conference proceedings (TREC 2005)*.

Yue, Y., Finley, T., Radlinski, F., and Joachims, T. (2007). A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*.

Zhang, J., Ackerman, M. S., and Adamic, L. (2007). Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on World Wide Web*.

Zhu, J., Song, D., and Ruger, S. (2007). The open university at trec 2006 enterprise track expert search task. In *Proceedings of the15th text retrieval conference proceedings (TREC 2005)*.

Zhu, J., Song, D., and Ruger, S. (2008). The open university at trec 2007 enterprise track. In *Proceedings of the 16th text retrieval conference proceedings (TREC 2007)*.