A Supervised Learning Approach to Entity Search

Guoping Hu[1], Jingjing Liu[2], Hang Li, Yunbo Cao, Jian-Yun Nie[3], and Jianfeng Gao

Microsoft Research Asia, Beijing, China
[1] iFly Speech Lab, University of Science and Technology of China, Hefei, China
[2] College of Information Science & Technology, Nankai University, Tianjin, China
[3] Département d'informatique et de recherche opérationnelle Université de Montréal

Abstract. In this paper we address the problem of entity search. Expert search and time search are used as examples. In entity search, given a query and an entity type, a search system returns a ranked list of entities in the type (e.g., person name, time expression) relevant to the query. Ranking is a key issue in entity search. In the literature, only expert search was studied and the use of co-occurrence was proposed. In general, many features may be useful for ranking in entity search. We propose using a linear model to combine the uses of different features and employing a supervised learning approach in training of the model. Experimental results on several data sets indicate that our method significantly outperforms the baseline method based solely on co-occurrences.

Keywords: Information Retrieval, Entity Search, Expert Search, Time Search, Supervised Learning.

1 Introduction

Most of the research on Information Retrieval (IR) focuses on search of documents. Despite the progress on document search, many specialized searches are still not well studied. This paper tries to address entity search, which provides search of specific types of information. More precisely, the user types a search query and designates the type of entity, and the system returns a ranked list of entities in the type (persons, times, places, organizations, or URLs) that are likely to be associated with the query. Entity search includes expert search, time search, place search, organization search, and URL search.

Entity search will be particularly useful at enterprise. People at enterprise are often interested in obtaining information of specific types. For example, an employee of a company may want to know the time of the next event concerning a product, the right person to contact for a problem, and so on.

Traditional search approach does not support entity search. For example, for a query looking for "date of IT Exhibition," each of the words will be used as a keyword, including "date." As a consequence, documents containing the keywords "date," "IT" (if not removed as a stop word) and "exhibition" will be retrieved. However, the documents or passages retrieved do not necessarily contain the required information, i.e. the date of the exhibition in this example.

There is an increasing interest in entity search in the research community. For example, expert search was investigated in several search engines and studies [2, 6, 7, 13, 14, 15, 21 and 22]. There was also a task on expert search at TREC 2005.

However, we observe that all the proposed methods relied on traditional IR techniques or simple features for ranking entities. For example, Craswell et al. [3] proposed using the co-occurrences of people and keywords in documents as evidence of association between them, and ranking people according to the strength of association.

One may consider dealing with the problem with Question Answering (QA). However, there is a striking difference between QA and entity search. Well formed questions are assumed to be inputted in QA, while most queries in entity search are just keywords. Therefore, a great number of the techniques used for QA cannot be adequately applied here.

In this paper, we aim to develop an appropriate approach for entity search. Ranking is a key issue in entity search. We propose a new ranking method that utilizes a variety of features in a linear combination model. Supervised learning is employed to train the model.

Two specific types of search are considered as examples in this paper: time search and people search. For time search, experiments have been carried out on two different collections: one from the intranet of Microsoft and the other from TREC Web and QA Tracks. For expert search, experiments have been performed with the TREC expert search data. The experimental results indicate our method performs significantly better than the baseline methods.

The rest of the paper is organized as follows. In Section 2 we introduce related work, and in Section 3 we explain the problem of entity search. In Section 4, we describe our proposed approach to entity search. Section 5 gives our experimental results. We conclude in Section 6.

2   Related Work

2.1   Traditional Search

Conventional Information Retrieval aims to identify documents or passages that may be 'relevant' to the query. The evidence used for the relevance judgment is the appearance of the query terms in the documents or passages. The query terms are usually weighted according to their occurrences in the documents or passages using models such as TF-IDF (e.g., [20]), BM25 (e.g., [19]), and Language Model (e.g., [17]). Usually these term weighting methods do not require labeled data for training. In that sense, the methods are unsupervised.

There is also a new trend in IR recently that manages to employ supervised learning methods for training ranking functions. Herbrich et al. [9] cast the IR problem as ordinal regression. They proposed a method for training the ordinal regression model on the basis of SVM. Gao et al. [8] proposed conducting discriminate training on a linear model for IR and they observed a significant improvement of accuracy on document retrieval by using the method. The success of the approach is mainly due to the following two factors: (1) The model provides the flexibility of incorporating an arbitrary number of features; (2) The model is trained in a supervised manner and thus has a better adaptation capability.

2.2   Entity Search

Entity search tries to identify entities strongly associated with query terms. There were several studies on entity search. The most studied type of entity was people (or expert).

Expert search was investigated in [2, 6, 7, 13, 14, 15, 21 and 22]. There was also a task on expert search at TREC 2005 [25]. All the existing methods for entity search only exploited simple features or traditional IR techniques for ranking. For example, Craswell et al. [3] proposed using co-occurrences between people and query words as evidence to rank people.

Many features may be useful for entity search, including new features that are not used in traditional IR. Therefore, an appropriate approach to entity search should easily incorporate new features. In this paper, we propose employing supervised learning to train an entity search model. This approach is chosen because of its flexibility and its capability of adaptation. To our knowledge, no previous study has explored the same approach for entity search.

2.3   Factoid Question Answering

Question Answering (QA) is a task that aims to provide the user with the correct answer to a question. Many QA systems were developed, including Webclopedia [10], NSIR [18], MultiText [4], MULDER [12], AskMSR [1], and the statistical QA system of IBM [11].

Factoid QA is the most studied subtask, which tries to answer questions about facts. It usually contains the following steps: (1) question type identification, (2) question expansion, (3) passage retrieval, (4) answer ranking, and (5) answer generation. In question type identification, the type of the question (that is also the type of the answer) is identified. In question expansion, the synonymous expressions of the question are created. In passage retrieval, relevant passages are retrieved with the content words in the question and its expansions. In answer ranking, the retrieved passages are ranked, and the potential answers within the passages are marked (in the identified question type). Finally, in answer generation, a single answer is generated from the top ranked passages.

People may take QA as an appropriate means for entity search. However, the assumptions for QA may not hold for entity search. First, QA heavily relies on NLP techniques to analyze the question. The basic assumption is that the question is a well formulated question. For entity search, we have to deal with queries instead of complete questions. That means that NLP would not help entity search. Second, users may be satisfied with one or several passages as search results as in entity search; getting a unified answer as in QA is more desirable, but not necessarily needed. Therefore, for entity search we do not necessarily need to employ the same methodologies used in QA.

3   Entity Search

The problem of entity search such as expert search and time search can be described as follows. The system maintains a collection of documents. When the user inputs a query (just like people usually do in search of documents) and designates a type, the system returns a ranked list of entities in the type based on the information in the documents. The entities are ranked based on the likelihood of being associated with the query. An entity is considered being associated with the query if there exists strong relationship between the query and the entity (e.g., a person is an expert on the

Fig. 1. An illustrative interface of entity search system

topic of the query). In practice, an association presented in a document can be bogus; however, we do not consider the problem in this paper.

A single user interface can be considered for entity search. There are check boxes on the UI, and each represents one type of entity (as shown in Figure 1). The user can designate the types of information to search by checking the corresponding boxes. The user can then input queries in the search box.

Figure 1 shows an example of expert search. When the user checks the box "expert search" and inputs the query "semantic web coordination," the system returns a list of people that are likely to be experts on the area. Each answer (person) is associated with supporting documents. The documents can help the user to figure out whether the answers are correct.

4   Our Method for Entity Search

In entity search, we first need to recognize the targeted entities in the documents. A simple method is used here: We utilize heuristic rules to conduct the entity recognition. For example, in time search, rules are created to identify time expressions such as "Nov. 17, 2004." Here, we do not consider conducting anaphoric resolution, for example, identifying the date of "yesterday" in a document.

For each identified entity, a passage is constructed. A passages is a window of fixed size around an identified entity; but it can also be delimited by the natural boundaries (tags in HTML documents such as <table>, <ol>, <p>, <ul>, <pre>, <li>, <dl>, <dt>, <tr>, <hr>). The entity in a passage is called the *center* of the passage. Around the center, features are extracted and used for the ranking of the passage with regard to the query.

The approach used in our study, which makes it different from most existing ones, is the utilization of a supervised learning model to combine different features. Our method is motivated by the following observations: 1) Features are usually extracted according to heuristics, and thus are different in nature. It is difficult to combine them in a traditional IR function like BM25. 2) Once new features are added, one has to re-tune the ranking function. This is difficult if the tuning is done manually. 3) Supervised learning is capable of finding the best combination of the features in an automatic manner, given a set of features and some training examples. The training process can be executed again, once new features are added. In this study, our goal is to develop a general method for entity search and thus we employ the supervised learning approach to perform the task.

4.1 Ranking Function

Two ranking functions are defined: one for passage and the other for entity. Our entity search method comprises of two steps: determining the top K passages and determining the top N entities from these passages.

Each passage retains a ranking score representing its association to the query. Let us denote it as $s(q, p)$. Suppose that the query q contains several keywords (content words). Then, $s(q, p)$ is determined according to the following equation:

$$s(q, p) = \sum_{w \in q \cap p} \varphi(w, p) \tag{1}$$

where w stands for a keyword appearing in both the query q and the passage p, and $\varphi(w, p)$ denotes the weight of w. We will explain the function $\varphi(w, p)$ in more details in the next subsection. We select the top K passages based on the $s(q, p)$ scores.

Once the K top passages are identified, a *weighted voting* is conducted to determine the top N entities, in which each passage votes for its own entity (in its center). Specifically, the ranking score of an entity e with respect to the query q is calculated as follows:

$$s(q, e) = \sum_{p \in P(e)} s(q, p) \tag{2}$$

where $s(q, e)$ is a score representing the association between query q and entity e, and $P(e)$ is the set of passages supporting e. The top N entities are those with the highest $s(q, e)$ scores.

4.2 Learning the Weighting for a Keyword

In conventional IR, the weight of a keyword is determined according to its frequency. For entity search, we believe that this weighting schema is not appropriate. Another mechanism should be adopted. For example, a keyword may be more important if it is closer to the entity (center) in the passage. Here, we define the general weighting schema of $\varphi(w, p)$ - the weight of query word w in a passage p - as a linear function of features. Each keyword w corresponds to a vector of m features denoted as $(x_{w,p,1}, ..., x_{w,p,m})$. Then $\varphi(w, p)$ is calculated as follows:

$$\varphi(w, p) = \sum_{j=1}^{m} c_j x_{w,p,j} \tag{3}$$

where c_j, $j = 1,...,m$ are weights of different features, which are trained by using Hill Climbing algorithm as described later.

4.3 Feature Sets

As features, we utilize information on the distance between center and keyword, the characteristics of keyword, the characteristics of passage, etc. We describe the features in this section.

Time Search. For time search, we define 35 features and Table 1 shows some examples of them. Here, we denote the entity as center, and the keyword as word. Some features take on numerical values, while some features are Boolean features.

Expert Search. For expert search, we define 15 features and Table 2 shows some examples of them. Here, we denote the entity in a passage as center, and the keyword as word too.

Table 1. Example of features for Time Search

Group	Type	Feature Description
1	Features of *word*	IDF of *word*
		Term Frequency (TF) of *word* in original document
		TF * IDF of *word*
		Distribution of *word* in original document, measured in entropy (c.f. [23])
		Part of speech of *word*
2	Features of passage	Bag of words in passage
		Size of passage
3	Features on position	Distance function $\exp(-\alpha \cdot d)$ where d is minimum distance between *center* and *word*, and α is parameter.
		Does *word* occur immediately preceding or following center?
		Does *word* occur in the same sentence or paragraph as *center*?
		Does *word* occur in the previous or next sentence of *center*?
4	Features on relation with other entity	Is there any other entity in the same sentence as *center*?
		Is there any other entity close to *word*?

4.4 Training

The key problem in ranking is to make a correct utilization of the features, i.e. a correct setting of the weights $(c_1,...,c_m)$ in equation (3). We resort to supervised learning to train the weights in the linear combination function. To do this, we need to utilize a set of

Table 2. Example of features for Expert Search

Group	Type	Feature Description
1	Features of *word*	Term Frequency (TF) of *word*
1	Features of *word*	TF * IDF of *word*
2	Features on position	Distance function $\exp(-\alpha \cdot d)$ where d is minimum distance between *center* and *word* and α is parameter.
2	Features on position	Does *word* occur immediately preceding or following *center*?
3	Features on metadata	Does *word* occur in the title of document?
4	Features on structure of document	Does *word* occur in the same section with *center?* (We parse the HTML document to get the section information)

training data. The training data contains a set of queries, a set of passages obtained as described above, and the relevance judgments provided by human judges.

Different algorithms can be employed in training of the weights $(c_1,...,c_m)$. In this study, we use the Hill-Climbing algorithm (e.g., [16]) because of its simplicity. The Hill-climbing algorithm is slightly modified so as to deal with over-tuning. This algorithm works as follows:

(1) Initialize each weight c_j as 0.0;
(2) For $j = 1$ to m
   (a) Set $t = c_j$
   (b) Try different weight values for t, conduct ranking using the training data and evaluate the performance of *entity ranking* in terms of Mean Average Precision
   (c) Record the highest performance achieved in b) and the corresponding weight value
   (d) Reset c_j as $t;$
(3) Record the highest performance achieved in step (2) and the corresponding weight, if the improvement of step (2) is larger than the threshold σ (this is to avoid over-tuning), then adopt the new weight (note that only one weight is updated here), and go to step (2). Otherwise, terminate the algorithm.

5   Experimental Results

5.1   Baseline Methods

Our method is compared with two baseline methods:

(1) **BM25:** In this method we calculate the BM25 score of the passage with respect to the query.
(2) **Distance:** In this method we calculate the following distance score:

$$\exp(-\alpha \cdot d(w,t)) \tag{4}$$

where w denotes a keyword and t denotes the *center*, d is the *minimum* distance between t and w, and α is a parameter.

The first baseline method is representative of the traditional IR approach. These baseline methods are chosen because they are commonly used in the previous studies. For example, Craswell et al.'s method based on co-occurrence [3] is similar to the BM25 method. The second one is similar to a typical QA approach [18].

For all the two methods, the top K passages are identified according to their passage ranking functions. Then two voting methods are used to further rank entities: simple voting or weighted voting. Simple voting means voting entities by the number of supported passages, and weighted voting is similar to that in equation (2).

5.2   Evaluation Measures

We use Mean Average Precision (MAP) and R-Precision [5] as the measures for evaluations of entity search.

5.3   Time Search

Data Sets. Our experiments on time search were carried out on two data sets:

1) The first one was created from the query logs of a search engine on the intranet of Microsoft in a time period (9/2004). First the queries containing the clue words "when," "schedule," "day," and "time" were collected, and then the clue words were removed from the queries. The remaining parts were used as pseudo queries time search. This set contains 100 queries (referred to as MS hereafter). The documents are from the same intranet.

2) The other query set was created from the temporal questions in the TREC QA Track (i.e., questions with "when," "in what time," etc). Again, stop words and time clues were removed from the queries. This query set contains 226 queries (referred to as TREC hereafter). The documents are those used in TREC Web Track.

For each query, each of the methods tested returned 100 answers. These answers were judged manually by 2 human evaluators. For 23% of the MS queries correct answer could not be found in the retrieved documents, and the number was 51% for TREC.

Time Expression Identification Experiment. As our method depends on the quality of time expression identification, we first conducted experiments on the time expression identification. 300 documents were randomly selected. A human annotator was asked to annotate all the time expressions within them. This enabled us to evaluate the time identification method. Table 3 shows the evaluation results. We see that in general our identification method obtains high accuracies. The lower precision for the MS data is due to the occurrences of confusing product names and programming codes in the data set (e.g., "Money 2002").

Time Expression Ranking Experiment. We conducted time search using our method and the two baselines. The top 100 answers with each method were used. We performed *4-fold cross validation*, i.e. 3/4 of the queries were used in training and 1/4

Table 3. Results of time expression identification method

Dataset	Annotated	Identified	Matched	Precision	Recall
MS	2,018	2,447	1,987	0.8120	0.9846
TREC	3,589	3,783	3,544	0.9368	0.9875

Table 4. Results of time expression ranking

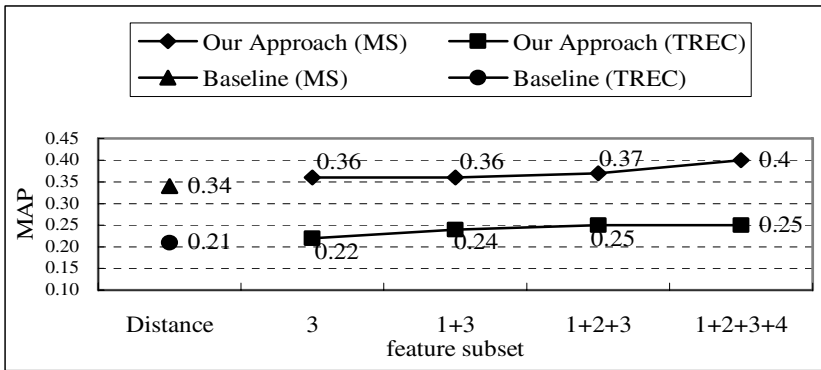| MS	MAP	R-Precision
BM25 (Simple Voting)	0.1483	0.1241
BM25 (Weighted Voting)	0.1559	0.1377
Distance (Simple Voting)	0.3291	0.2880
Distance (Weighted Voting)	0.3440	0.3070
Our method	**0.3952**	**0.3659**
TREC	MAP	R-Precision
BM25 (Simple Voting)	0.1232	0.1012
BM25 (Weighted Voting)	0.1277	0.1158
Distance (Simple Voting)	0.1951	0.1760
Distance (Weighted Voting)	0.2070	0.1863
Our method	**0.2454**	**0.2234**

Fig. 2. Ranking results with different feature subsets for passage scoring learning

for testing. Table 4 shows the time search results of the different methods in terms of MAP and R-Precision.

We can observe that Distance performs better than BM25. This shows that the traditional IR method that does not incorporate a specific treatment of entities is not sufficient for entity search. We can also observe that weighted voting performs better than simple voting. However, the differences are not large. Comparing our method

with the baseline methods, we can see that our learning method outperforms both baseline methods with quite large margins. One may notice that the overall accuracies in terms of MAP and R-Precision are not very high. An important reason is that 23% of the MS queries and 51% of the TREC queries do not have correct answers.

In order to analyze the contribution of each feature subset (representing different aspects) in our learning based method, we used different subsets of the features in our ranking functions. The feature subsets 1, 2, 3 and 4 in Figure 2 correspond to the four feature aspects described in Section 4.3. From Figure 2, it can be seen that when more features are used the search results can be improved. This result validates our hypothesis that more features should be used in entity search. It also shows that the supervised learning method we propose is capable of combine different features so as to take advantage of each of them.

5.4 Expert Search

Data Set. In our expert search experiments, we used the data set in the expert search task of enterprise search track at TREC 2005. The document collection was crawled from the public W3C [24] sites in June 2004, which contains 331,307 web pages, including specifications, email discussion, wiki pages and logs of source control. The ground truth on expert search was obtained from an existing database of W3C working groups. For each query (the name of the working group), a list of people (the group members) is associated as experts on the topic. There are in total 1,092 members in all the groups. (For details of the TREC data, see [25]).
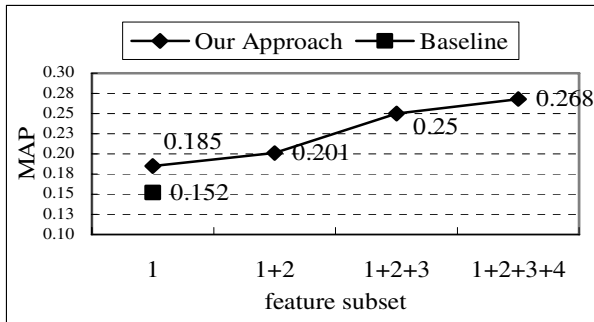
Personal Name Identification. Heuristic rules were used to identify personal names in the documents. In order to evaluate the accuracy of this process, 500 documents from the W3C document collection were randomly selected. All the personal names from these documents were manually checked by a human annotator. The evaluation result shows that our name identification method can work well: the precision and recall are 100% and 90%, respectively. The process missed some names mainly because some irregular variants of names exist in the corpus (e.g. "danc" for "Dan Connolly"). In general, our name identification process is satisfactory.

Expert Ranking Experiment. In the official tests in TREC, two sets of queries were provided: a set of 10 training topics and another set of 50 topics for testing. However, we notice that quite large differences between the training and testing queries exist: All the queries in the training set are single phrases, while some queries in the test data are combinations of several phrases. Therefore, we only used the 50 test queries in our experiment, and conducted 10-fold cross-validation in our evaluation. The results reported below are the averaged results over 10 trials.

Table 5 shows the results on expert search for our method and the baseline methods. Different from time search, we do not observe a large difference between the BM25 method and the Distance method: both baseline methods perform quite poorly. In comparison, our method based on supervised learning outperforms them significantly. This confirms again that our method is more suitable for entity search.

Table 5. Results of the baseline methods and our method

Methods	MAP	R-Precision
BM25 (Simple Voting)	0.1516	0.1623
BM25 (Weighted Voting)	0.1231	0.1352
Distance (Simple Voting)	0.1430	0.1625
Distance (Weighted Voting)	0.1330	0.1644
Our method	0.2684	0.3190

Fig. 3. Ranking results with different feature subsets for passage scoring learning in expert search

In order to see the contributions of different features, we ran our method with different subsets of features. From Figure 3, we can see that the more the features used, the better the performance achieved. This is similar to the observation on time search. Therefore, we have more evidence to say that it is advantageous to incorporate more features in entity search and employ a supervised learning method to appropriately combine the uses of the features.

6 Conclusion

Entity search such as expert search and time search is useful in many search scenarios, particularly in enterprise. Methods based on the use of co-occurrence have been proposed in the literature for expert search. This paper explores a new approach for entity search on the basis of supervised learning. Specially, it makes use of a linear combination model for ranking of entities, which incorporates many different useful features.

We have applied our method to two of the entity searches – time search and expert search in this paper. Experimental results show that the proposed method performs significantly better than the baseline methods solely using co-occurrence or distance.

The main contributions of this work are (1) a proposal of a supervised learning approach to entity search and an empirical verification of the effectiveness of the approach; and (2) identification of some useful features for time search and expert search.

Time search and expert search are just the first entity searches we have investigated. Our long term goal is to construct a search system in which all the commonly required entity searches are developed effectively. Such a system would offer great facilities for people to find their required types of information.

References

1. Eric Brill, Susan Dumais and Michele Banko, *An Analysis of the AskMSR Question-Answering System*, EMNLP 2002

2. Christopher S. Campbell, Paul P. Maglio, Alex Cozzi, Byron Dom, *Expertise Identification using Email Communications*, CILM'03, 2003

3. N. Craswell, D. Hawking, A. M. Vercoustre and P. Wilkins, *P@NOPTIC Expert: Searching for Experts not just for Documents*. In Ausweb, 2001.

4. Gordon V. Cormack, Charles L. A. Clarke, D. I. E. Kisman, Christopher R. Palmer, *Fast Automatic Passage Scoring (MultiText Experiments for TREC-8)*. TREC 1999

5. Gordon V. Cormack and Thomas R. Lynam, *Statistical Precision of Information Retrieval Evaluation*, SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.

6. S. Deerwester, S. T. Dumais, G. W. Fumas, T. K. Landauer, and R. Harshman, *Indexing by Latent Structure Analysis*, Journal of the American Society for Information Sciences. ACM Press 391-407, 1990.

7. B. Dom, I. Eiron, A. Cozzi and Z. Yi, *Graph-Based Ranking Algorithms for E-mail Expertise Analysis*, in Proc. of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, 2003.

8. Jianfeng Gao, Haoliang Qi, Xinsong Xia, and Jian-Yun Nie. 2005. *Linear discriminant model for information retrieval*. In: SIGIR2005.

9. R. Herbrich, T. Graepel and K. Obermayer. 2000. *Large margin rank boundaries for ordinal regression*. Advances in Large Margin Classifiers, pp. 115-132. MIT Press, Cambridge, MA.

10. Eduard H. Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, Chin-Yew Lin, *Question Answering in Webclopedia*. TREC 2000

11. Abraham Ittycheriah, Salim Roukos, *IBM's Statistical Question Answering System-TREC 11*. TREC 2002

12. Cody C. T. Kwok, Oren Etzioni, Daniel S. Weld, *Scaling question answering to the Web*. WWW-2001: 150-161

13. M. E. Maron, S. Curry, and P. Thompson. *An inductive search system: Theory, design and implementation*. IEEE Transaction on Systems, Man and Cybernetics, 16(1):21–28, 1986.

14. Dave Mattox, Mark Maybury, Daryl Morey, *Enterprise Expert and Knowledge Discovery*, Proceedings of the HCI International '99

15. D. W. McDonald, *Evaluating Expertise Recommendations*. In Proc. of the ACM 2001 international conference on Supporting Group Work (GROUP'01), Boulder, CO, 2001.

16. William Morgan, Warren Greiff, John Henderson, *Direct Maximization of Average Precision by Hill-Climbing, with a Comparison to a Maximum Entropy Approach,* HLT-NAACL 2004, pp. 93-96

17. Jay M. Ponte, W. Bruce Croft, *A Language Modeling Approach to Information Retrieval*. SIGIR1998: 275-281

18. Dragomir R. Radev, Weiguo Fan, Hong Qi, Harris Wu, Amardeep Grewal, *Probabilistic question answering on the web*. WWW 2002: 408-419
19. Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Mike Gatford, A. Payne, *Okapi at TREC-4*. TREC 1995
20. Gerard Salton, James Allan, Chris Buckley, *Approaches to Passage Retrieval in Full Text Information Systems*. SIGIR 1993: 49-58
21. Mayssam Sayyadian, Azadeh Shakery, AnHai Doan, ChengXiang Zhai, *Toward Entity Retrieval over Structured and Text Data*, WIRD'04, the first Workshop on the Integration of Information Retrieval and Databases (WIRD'04), 2004
22. L.A. Steer and K.E. Lochbaum, *An Expert/Expert Locating System Based on Automatic Representation of Semantic Structure*, in Proc. of the Fourth IEEE Conference on Artificial Intelligence Applications, 1988.
23. Li Zhang, Yue Pan, Tong Zhang, *Focused named entity recognition using machine learning*. SIGIR 2004: 281-288
24. World Web Consortium (W3C), http://w3.org
25. TREC 2005, http://trec.nist.gov/tracks.html