Research on Expert Search at Enterprise Track of TREC 2006

Shenghua Bao[1], Huizhong Duan[2], Qi Zhou[3], Miao Xiong[4], Yunbo Cao[5] and Yong Yu[6]

APEX Data & Knowledge Management Lab,
Shanghai Jiao Tong University, Minhang District
Shanghai, P.R.China, 200240
{shhbao[1],summer[2], jackson[3],xiongmiao[4], yyu[6]}
@apex.sjtu.edu.cn

Microsoft Research Asia
5F Sigma Center, No.49 Zhichun Road, Haidian
Beijing, P.R.China, 100080
yunbo.cao[5]@microsoft.com

1. INTRODUCTION

This year, we (SJTU team) participated in the Expert Search task at the Enterprise Track. Last year, two of the members participated in the Expert Search task (MSRA team) [1]. This document reports our new experimental results on the expert search of TREC 2006.

In this research, we propose a new evidence-oriented framework to expert search. Here, the evidence is defined as a quadruple, <Query, Expert, Relation, Document>. Each quadruple denotes that a "Query" and an "Expert", with a certain "Relation" between them, are found in a specific "Document". In the proposed framework, the task of Expert Search can be accomplished in three steps, namely, 1) evidence extraction, 2) evidence quality evaluation, and 3) evidence merging. Thus, our experiments include the following items. We will explain them in detail later in the following sections.

1) **Evidence extraction**: In the participation of last year [1], we have proposed as the evidences several effective relations, such as window-based co-occurrence, block-based co-occurrence, author-title co-occurrence, etc. This year, we explore two new relations, namely semantic-block-based co-occurrence and improved block-based co-occurrence.

2) **Evidence quality evaluation**: The quality of each kind of evidence (or relation) varies in terms of contributing to experts ranking. We, in the report, consider the following factors to evaluate the quality of the evidences,

- Relation-type quality: different relations should carry on different confidence in indicating the strength of the connection between the expert and query. In our system, the improved block-based co-occurrence relation is given the highest confidence.
- Query-matching quality: the query can match its occurrence in the documents in various ways. One direct way is called phrase matching representing the exact matching. However, the exact matching cannot happen always, especially for the long queries. For the case, we try several relaxed matching methods, such as bi-gram matching, proximity matching, fuzzy matching and stemming matching. We will detail the matching methods later. Obviously, different matching methods represent different qualities.
- Expert-matching quality: an expert candidate can occur in the documents in various ways. The most confident occurrence should be the ones in full name or email address. The other ways can include last name only, last name plus initial of first name [1].
- Context quality: the quality of evidence also depends on the quality of the context in which it is found. More specifically, as the context, we consider the documents. It's the quality of a document supporting an evidence consists of two parts: a) static quality -- indicating the authority of the document, e.g. corpus type, PageRank [4] etc.; b) adaptive quality -- it varies with the queries, e.g., the relevance score between the document and the query.

3) **Evidence merging**: we compare two methods for evidence merging.

- Simple-Merging method: the scores associated to the same expert is linearly combined together to reach a final evaluation on the expert.
- Smoothed-Merging method: the method strengthens the simple-merging method by smoothing it using cluster based re-ranking. Before fed into the smoothed-merging method, the simple-merging score is mapped into its logarithm value due to the fact that the score of expert varies sharply.

The experimental results show that the new explored evidences and the evaluation of evidence quality can really improve the expert search significantly. The results also show that Smoothed-Merging method can boost the performance further on the basis of Simple-Merging method.

All the results except those in Section 6 are obtained from the experiments on last year's test set (EX01—EX50). By incrementally adding the components considered above, we are able to achieve the best result on the test data set. We then apply the experience to the test queries (EX51-EX105) of this year as described in Section 6.

2. EVIDENCE ORIENTED FRAMWORK FOR EXPERT SEARCH

Figure 1 shows the overview of the evidence based framework for the expert search task. The framework begins with processing the given query *Q*, including the removal of stop-words and the transformation into several other forms. The set of transformed queries is used for evidence extraction, in which we explore as the evidences several effective co-occurrence relations. The outputs of the evidence extraction are quadruples as 'Quad <Query, Expert, Relation, Document>'. These quadruples then are fed into the step of quality evaluation and associated with quality scores as <Quad, Score>. As the final step, the evidence merging groups all the quadruples by query and expert and fuses the scores with a certain merging method. The result presented to users is a list of experts, each consisting of a name, a score and a list of supporting evidences.

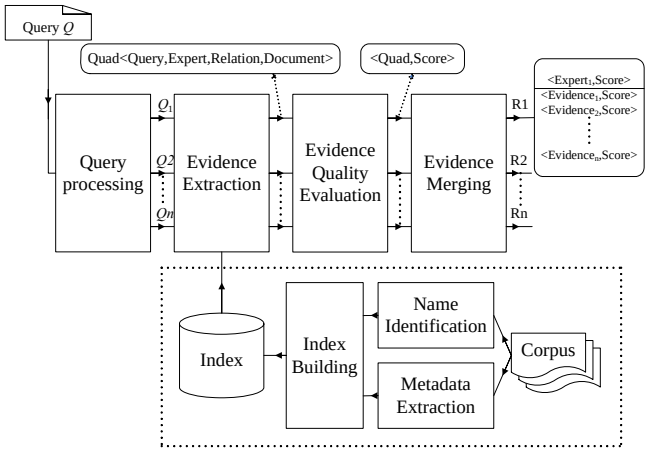

Figure.1 Overview of Evidence Oriented Framework

3. OFFLINE PROCESSING

3.1 Personal Name Identification

The identification method used this year is similar to the one used in last year [1]. The only difference is the introduction of email alias and new email inference.

Table 1 shows a sample of person name identification. Full name and email name can be directly obtained from given expert list. Rest of masks is our augmented expert identification masks. Percentage is a statistic of the name appears in the W3C corpus [5] and ambiguity reflects the probability of whether a name under that mask will be shared by more than one person.

Table 1. Proportion and Ambiguity of Various Person Names

Mask	Sample	Rate	Ambiguity
Full Name	Ritu Raj Tiwari Tiwari, Ritu Raj	48.2%	0.0%
Email Name	rtiwari@nuance.com	20.1%	0.0%
Combined Name	Tiwari, Ritu R R R Tiwari	4.2%	39.92%
Abbreviated Name	Ritu Raj Ritu	21.2%	48.90%
Short Name	RRT	0.7%	63.96%
Alias, New Mail	Ritiwari rtiwari@hotmail.com	7%	0.46%

Introducing possible masks of expert may bring some noise as the cost. Homonymy (more than one expert share one name) is the biggest problem: 'David' from given name mask is ambiguous for it has been shared by 28 experts in expert list. Thus, for each person in the list, we matched it against the documents and located all the positions of it using the heuristic rules as Table 2. We could then index the information with regard to each of the persons by Lucene [3].

Table 2: Heuristic Rules for Identifying Personal Names

1) Every occurrence of a candidate's email address is normalized to the appropriate candidate_id.
2) Every occurrence of a candidate's full_name is normalized to the appropriate candidate_id if there is not ambiguity; otherwise, the occurrence is normalized to the candidate_id of the most frequent candidate with that full_name.
3) Every occurrence of combined name, abbreviated name, and email alias is normalized to the appropriate candidate_id if there is not ambiguity; otherwise, the occurrence may be normalized to the candidate_id of a candidate whose full name has also appeared in the document.

4) All the personal occurrences other than those covered by Heuristic 1) ~ 3) are ignored.

3.2 Metadata Extraction

Following the participation of last year, we explore more on the use of metadata for expert ranking. Besides author, title, and section-based block, we further consider the extraction of the reference block and mailing archives.

Reference blocks are usually composed of publishers and authors. The reference blocks can serve as a reliable source which connects the topics and expert candidates in a more tight way. It might be argued that the metadata and its extraction are specialized for the data set of w3c sites. Actually, we can find other examples such as wikipedia [7].

We also extracted the sender, receiver, and title parts from email archives. Duplication in email achieves such as 'By Author', 'By Topic', or 'By Date' is also detected and removed in our extraction.

4. ONLINE PROCESSING

4.1 Query Processing

In the query processing, we form phrase queries by removing the stop-words and then transform them into four kinds of variations.

Bi-gram query: a set of queries, each in the set representing a word bi-gram extracted from the original phrase query.

Proximity query: a query each term of which appear as neighborhood within a window of specified size. Matching the query of this kind should leverage the efficient implementation of the indexing.

Fuzzy query: a set of queries, each in the set resembles the original query in appearance. For instance, query "mereology" may find "methology" or "ontology" as its fuzzy queries.

Stemming query: a query obtained by stemming each word in the original phrase query. The query should be used with an index of stemmed documents together.

4.2 Evidence Relations

4.2.1 Evidence Relations Extraction

In the evidence extraction, we explore five most effective relations including some reported in [1]. They are window-based co-occurrence relation, section-constrained co-occurrence relation, block-based co-occurrence relation, metadata-based co-occurrence relation and semantic-block-based co-occurrence relation. Among them, semantic-block-based co-occurrence relation is the new founding in the report. As for the block-based relation, we explore some improvements on it. As for the other three, we take the same approach as our work of last year [1].

Semantic-block-based relation: expert and query co-occur in a semantic block. A semantic block is a piece of information extracted from the reference block or email archive, etc.

Improved block-based relation: this year, we build two kinds of section-trees for block-based relation extraction, namely query section-tree and expert section-tree. The query section-tree is similar to the one in [1] but two additional constraints: a) restricting the match depth of the tree-section model by a threshold. (tree-depth constraint) b) removing the reference block from the section-tree matching (reference block removal). Expert section-tree is opposite to the query section-tree in which the expert appearing in the <h1> to <h6> while the query appearing in the text block. The expert section-tree relation is extremely useful when a page is an introduction on some experts.

By exploring these five relations, we are able to find the most useful evidences for all the candidate experts. Each expert and evidence pair, together with their query and relation type, is formed into a quadruple for the evidence quality evaluation step.

4.2.2 The Base Model for Expert Search

Now, we are facing the problem of how to calculate the relevance between the expert and the given query, given a single piece of evidence <*q*, *e*, *r*, *d*>. As described in [1], the relevance value is calculated using the language model with smoothing, as described below:

$$S_{base}(q,e,r,d) = \mu \frac{freq(e,d,r)}{L(d,r)} + (1-\mu) \sum_{d':e \in d'} \frac{freq(e,d',r)}{L(d',r)} / df_e \quad \textbf{(1)}$$

Where *freq(e,d,r)* is the frequency of expert *e* matched by relation *r* in document *d*, and *L(d,r)* is the frequency of all the experts matched by relation *r* in *d*. df_e is the document frequency of expert *e*. We use Dirichlet prior in smoothing of parameter μ:

$$\mu = \frac{L(d,r)}{L(d,r)+K} \quad \textbf{(2)}$$

Where *K* is the average term frequency of all experts in the collection.

4.3 Evidence Qualities

4.3.1 Quality Types

The quality of each quadruple <Query, Expert, Relation, Document> consists of four aspects, namely relation-type quality, expert-matching quality, query-matching quality, and context quality. The overall quality score is the combination of the scores from these four aspects. Let <*q*, *e*, *r*, *d*> be a specific extracted evidence and the detailed evaluation procedures are described as follows.

Relation-type quality.

Different relations should carry on different confidence in indicating the strength of the connection between the expert and query. In our system, the improved block-based co-occurrence relation is given the highest confidence. The final quality score Q_r of relation r is:

$$Q_r = W_r D(r,q,e) \quad \textbf{(3)}$$

where W_r is the weight of relation type r, $D(r, q, e)$ is the distance factor of the query q and expert e within the relation r. The distance factor is calculated as,

$$D(r,q,e) = \begin{cases} \dfrac{100}{dis(e,q)+1} & if \quad r = \text{window constrained relation} \\ 1 & else \end{cases} \quad \textbf{(4)}$$

where *dis(e, q)* is the distance from the expert e to the query q.

Expert-matching quality.

An expert candidate can occur in the documents in various ways. The most confident occurrence should be the ones in full name or email address. The other ways can include last name only, last name plus initial of first name [1]. Thus, the action of rejecting or accepting a person from his/her mask (the surface expression of a person in the text) is not simply a boolean decision, but a probabilistic one with a reliability weight Q_e. For example, the weight Q_e for full name is 1 while that for abbreviated name is 0.73.

Query-matching quality.

As mentioned before, the query preprocessing step outputs several variations of the original phrase queries. The different forms of queries correspond to different query matching methods. Obviously, different matching methods represent different qualities. Thus, different query matching should be associated with different weights. The highest weight is for phrase query, then proximity query and bi-gram query.

We further note that, with bi-gram query, it is essential to distinguish the importance of each bi-gram in a set of bi-gram queries. For example, both "css test" and "test suite" are bi-gram queries constructed from the query "css test suite"; however, obviously that the former should be weighed more for it carries more information. To model that, we use the number of the returned documents to refine the query weight.

Thus, the query-matching quality score reflecting two factors above is,

$$Q_q = \begin{cases} W(t_q)\dfrac{MAX_{PQ(q')=PQ(q)}(df_{q'})}{df_q} & if \quad t_q = \text{bi - gram matching} \\ W(t_q) & else \end{cases} \quad \textbf{(5)}$$

where $W(t_q)$ is the weight for different query type, $PQ(q)$ refers to the corresponding phrase query of a given query q. df_q is the number of the returned document of the bi-gram query q. For the queries other than the bi-gram queries, constants are used as the weights.

Context quality:

The quality of an evidence also depends on the quality of the context in which it is found. Here we focus on considering the context of document, called 'document context'. The document context can affect the credibility of the evidence in two-folds:

1) Static quality: indicating the authority of the document, e.g. corpus type, PageRank [4] etc. In the experiments, we assign the highest weight to the corpus type of email. The PageRank of each document is obtained by analyzing the in-links and out-links. The static quality related to PageRank and corpus type are denoted as $Q_{PR}(d)$ and $Q_{CP}(d)$ respectively.
2) Dynamic quality. By "dynamic", we mean the quality score varies with different queries q. We denote the dynamic context quality as $Q_{DY}(d,q)$, which is actually the document relevance score returned by BM25[6] search .

Given the static and dynamic quality score, we define the context quality score as:

$$Q_d = Q_{PR}(d)Q_{CP}(d)Q_{DY}(q,d) \quad \textbf{(6)}$$

4.3.2 Quality Based Model for Expert Search

The entire quality of the quadruple <q, e, r, d> is the combination of the relation-type quality, expert-matching quality, query-matching quality, and context quality. Thus, the entire score used to ranking an expert candidate is defined as:

$$S_{quality}(q,e,r,d) = Q_q Q_e Q_r Q_d S_{base}(q,e,r,d) \quad \textbf{(7)}$$

Finally each quadruple corresponds to a pair <Quad,Score>.

4.4 Evidence Merging

For evidence mergence, we propose two methods, namely simple-merging method and smoothed-merging method.

4.4.1 Simple Merging method

By simple-merging (linear addition) here, we assume that the ranking score of an expert can be acquired by summing up together all scores of the supporting evidences. Thus we calculate experts' scores by aggregating the scores from all the <Quad,Score> pairs. The calculation is:

$$S_{simple}(e) = \sum_q \sum_r \sum_d S_{quality}(q,e,r,d) \quad \textbf{(8)}$$

where $S_{quality}$ is the score whose quadruple contains the expert.

4.4.2 Smoothed Merging method

We note that experts on the same research area often co-occur in the same context in a document. Thus we use a cluster-based approach to re-rank the returned experts for a certain query. The user clusters are generated in two steps:

1) Build each user's profile by extracting and merging surrounding text of his/her every occurrence.

2) Divide the expert's profiles into 20 clusters using K-means.

Then we re-calculate the expert's score as follows:

$$S_{smoothed}(e) = \lambda S_{\ln}(e) + (1-\lambda)\sum_{e' \in C} S_{\ln}(e') / |C|, \quad \textbf{(9)}$$

where C is the cluster that expert e belongs to, $|C|$ is the count of experts in cluster C. In experiment we use $\lambda = 0.5$. Due to Simple Merging method has a vital drawback for smoothing, that is, the score an expert acquires increases drastically as its evidence count increases. Thus the final score of an expert may be very high, and scores differ greatly from expert to expert. To certain extent, this is incompatible with the real world experience. Especially, this is infeasible for cluster based re-ranking, for the average score of a cluster may fluctuate at a large range. To complement this drawback, we use Logarithm operation to smooth expert score:

$$S_{\ln}(e) = Ln(S_{simple}(e)) \quad \textbf{(10)}$$

5. EXPERIMENTAL RESULT

To evaluate the effectiveness of the proposed framework, we construct the baseline as follows: 1) all the effective relations used in TREC 2005 are extracted; 2) the extracted evidences are merged with only relation type quality consideration. The best relation type quality is assigned to the baseline system.

5.1 Evaluation of Multiple Query Types

In our experiment, the fuzzy and stemming queries are put into use only when the returned experts are less than a threshold for the prior queries and the threshold is set to 100. Table 3 shows the result of expert search by using the multiple query types. The performance increases stably by adding new queries.

Table 3. Multiple Queries for Expert Search

	MAP	Bpref	Prec@10
Baseline	0.2379	0.4113	0.3380
+ Bi-gram query	0.2503	0.4970	0.3620
+ Proximity query	0.2556	0.5126	0.3680
+ Fuzzy, Stemming query	0.2564	0.5245	0.3700

5.2 Evaluation of Revised Relation Extraction

Based on the best result achieved in last subsection, we further evaluated the effectiveness of the new proposed semantic-block-based relation and the improved-block-based relation. Table 4 shows the improvements.

Table 4. Revised Expert Relation Extraction for Expert Search

	MAP	Bpref	Prec@10
+ Semantic-block	0.2584	0.5319	0.3720
+ Improved-block	0.2585	0.5317	0.3740

5.3 Evaluation of Evidence Quality

The performance of expert search can be further improved by considering the evidence qualities. Table 5 shows the results by considering different qualities incrementally.

Table 5. Evidence Qualities Consideration for Expert Search

	MAP	Bpref	Prec@10
+ Expert-matching quality	0.2628	0.5321	0.3714
+ Context dynamic quality (similarity)	0.2711	0.5358	0.3720
+ Context static quality (PageRank)	0.2749	0.5427	0.3840
+ Context static quality (corpus weighting)	0.2755	0.5424	0.3880

5.4 Evaluation of Evidence Merging

All the experimental results shown in the previous sections are based on simple merging. Table 6 shows the comparison between the simple merging and smoothed merging. The smoothed merging generates the best results (0.2941) for our evidence oriented framework which outperforms last years best result 0.2749 (THUENT0505) [2] by 6.984%.

Table 6. Simpe Merging vs. Smoothed Merging

	MAP	Bpref	Prec@10
Simple Merging	0.2755	0.5424	0.3880
Smoothed Merging	0.2941	0.5755	0.4380

6. SUBMITTED RUNS

Table 7 shows the performance of the four runs on this

year's test queries (EX51—EX105). All the four runs were based on the evidence based framework described in Section 4. The major differences between the four runs are described in Table 8.

Table 7. The performance four submitted runs

	MAP	Bpref	Prec@10
SJTU01	0.5829	0.5835	0.6878
SJTU02	0.5860	0.5863	0.6918
SJTU03	0.5851	0.5849	0.6776
SJTU04	0.5947	0.5913	0.7041

Table 8. Detail description of four submitted runs

Runs	Query Processing	Evidence Extraction	Quality Evaluation	Merging Method
Base	+ Phrase query + Bi-gram query + Proximity query + Fuzzy query + Stemming query	+ Window-based + Section-based + Title-Author + Semantic-block + Improved-block	+ Relation-type quality + Query-matching quality + Expert-matching quality + Context quality (Similarity)	
SJTU01	Same as Base	Same as Base	+ Context quality (PageRank)	Simple Merging
SJTU02	Same as Base	Same as Base	+ Context quality (PageRank) + Context quality (Corpus weighting)	Simple Merging
SJTU03	Same as Base	Same as Base	+ Context quality (Corpus weighting)	Simple Merging
SJTU04	Same as Base	Same as Base	+ Context quality (PageRank) + Context quality (Corpus weighting)	Smoothed Merging

7. REFERENCES

[1] Y. Cao, J. Liu and S. Bao and H. Li: Research on Expert Search at Enterprise Track of TREC 2005. In: *Proceedings of 14th Text Retrieval Conference (TREC 2005),* 2005.

[2] N. Craswell, A.P. de Vries, and I. Soboroff: Overview of the TREC 2005 Enterprise Track. In: *Proceedings of 14th Text Retrieval Conference (TREC 2005),* 2005.

[3] http://lucene.apache.org/.

[4] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. *Technical report*, Stanford Digital Library Technologies Project, 1998. Paper SIDL-WP-1999-0120 (version of 11/11/1999).

[5] http://research.microsoft.com/users/nickcr/w3c-summary.html.

[6] Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gull, A., Lau, M.: Okapi at TREC. In:*Text REtrieval Conference.*

[7] http://en.wikipedia.org