Solving the Enterprise TREC Task with Probabilistic Data Models

Jan Frederik Forst
Department of Computer Science
Queen Mary, University of London
E1 4NS London, UK
frederik@dcs.qmul.ac.uk

Anastasios Tombros
Department of Computer Science
Queen Mary, University of London
E1 4NS London, UK
tassos@dcs.qmul.ac.uk

Thomas Rölleke
Department of Computer Science
Queen Mary, University of London
E1 4NS London, UK
thor@dcs.qmul.ac.uk

ABSTRACT

Expert identification has become an important information retrieval task. We present and investigate a number of approaches for identifying an expert. Different approaches are based on exploiting the structure of documents in the knowledge base. Furthermore, our system highlights the integration of database technology with information retrieval (DB+IR).

1. INTRODUCTION

It is vitally important for businesses and organisations to acquire methods and technologies to organise data and retrieve information from data collections, especially with regard to the increasing amounts of data stored in corporate networks. Typically, corporate information is stored in large relational databases, where retrieval is carried out using SQL or other, related, languages.

One approach at organising data that has gained prominence more recently is that of *expert finding*. Large organisations spend considerable efforts improving their collaborative infrastructures: intranets and shared document authoring are examples for providing IT support for collaborative work. In such organisations, or in loosely coupled collaborative networks, the requirement for effective identification of experts frequently occurs.

In this paper, we place special emphasis on the integration of classical DB technologies with the more IR oriented aspects of expert identification. To this end, we used our retrieval framework, HySpirit [9], which integrates DB concepts, such as relational data representation and querying languages with probabilistic aspects commonly found in IR. HySpirit is based on results reported in [4] and [8], and has been improved steadily over recent years. This report will highlight the integration of probabilistic information retrieval models with SQL and Datalog expressions commonly found in the DBMS realm. The integration of such database technologies allows retrieval strategies to be implemented using very few lines of code, and enables retrieval systems to benefit from the efficiency of database management systems.

2. BACKGROUND

A good overview of the particularities and problems encountered in conducting information retrieval in an enterprise environment is given by Hawking [5]. Among the challenges mentioned are the need to handle heterogeneous data sources such as databases, web-servers, email repositories and content management systems, to allow for fine-grained access-control rights for different users, to support multiple document formats and to integrate all the results in a seamless fashion.

In earlier work, Hawking *et al.* introduced their expert finding system, P@NOPTIC Expert [2]. This system builds fairly simple expert profiles by concatenating all documents published by a person into one long document. Then, in order to retrieve a ranked list of experts for a query, classical document retrieval on all those documents is performed for that query.

Expert finding approaches based on clustering were developed by Foner, and Reichling *et al.*. In Foner's *Yenta*

system [3], users, and their interests, are represented by agents. Agents representing users with similar interest form clusters in the system, which can be used to identify user groups with common interests, or to find an expert for a given problem.

Similarly, Reichling *et al.* use expert profiles to match users in their system [6] . Their architecture is based on a learning platform; user profiles were built from information directly entered by the users (such as education, current occupation, past experience), and information derived from the browsing activities of users, i.e. documents viewed by users on the learning platform.

Yimam-Seid and Kobsa present an architecture, DEMOIR [11], which covers several aspects of the process of generating and querying expert profiles. Their system covers the three initial stages of building and representing expert-profiles, namely evidence source recognition - using either explicit sources, such as the experts themselves, or implicit sources (e.g. documents authored by expert-candidates), expertise indicator extraction - techniques to extract the salient information from the evidence sources, and expertise models - representations of extracted salient features.

Apart from research and experimentation in academia, some commercial systems have also developed expert profiles from implicit data sources. Autonomy[1] has included in its *IDOL Server* the ability to derive user profiles based on the documents that users access and submit on the company intranet. Both *KnowledgeMail* by Tacit[2] and *Discovery Server* by Lotus[3] build user profiles by scanning employees' emails.

3. KNOWLEDGE REPRESENTATION

In this section, we present an overview of how the indexed document collection and queries are stored in our system, how term- and document-frequencies (collection frequency) are calculated, and how associations between documents and expert-candidates are modelled.

3.1 Representation of documents and queries

Documents in our system are represented by relations, detailing the content and structure of the documents. Modelling follows the object-relational representation outlined in [7]. These relations are generated in an indexing stage, which also performs additional pre-processing steps such as stop-word removal and stemming. While the original TREC Enterprise track (ETREC) collection consists of HTML documents, we did not make use of the special markup as such, however, the documents' <*body*>-tags (or <*pre id="body"*>) were used to distinguish the actual document-content from the additional surrounding information; only terms occurring inside those tags were indexed, and the tags themselves were removed.

While the object-relational representation retains the original structure of documents, to allow for more complex query formulations, for our experiments only the *term*-relation was used. The *term*-relation contains <*term,context*> tuples, where *context* is the context in which a term occurs, here: documents.

To show this in more detail, consider the following example. For this sample document (*lists-054-0017908*)

<*pre id="body"*>
Hi Folks,
This is a message that at times I thought I might never get to send.
[...]

Thank you.
[...]

** I tried to sync up with Danbri but have failed to get in touch, so whilst*
I can't speak for him, I hope this message reflects his views as well.
</*pre*>

the *term*-relation would contain the following tuples:

0.459596	hi	lists-054-0017908
0.459596	folk	lists-054-0017908
0.459596	tim	lists-054-0017908
0.459596	thought	lists-054-0017908
0.459596	might	lists-054-0017908
[...]	[...]	[...]

The first column of the table gives the probability that a tuple is contained in the relation, a feature which we use to represent information which can be used similar to term-frequency (*tf*) in classical information retrieval. This is explained in more detail below. Note that tuple-probabilites are not an argument of the relation, but are used internally to derive probabilities under relational operations.

Queries are represented in a similar way, with the context being defined simply as "this". In addition,

[1]www.autonomy.com
[2]www.tacit.com
[3]following the acquisition of Lotus by IBM, Discovery Server was discontinued.

query-terms are weighted with the collection-frequency (idf-frequency), rather than the term-frequency (tf-frequency).

3.2 Term- and Document-frequencies

Similar to the way classical IR systems define *tf*- and *idf*-frequencies, we derive from our relations probabilities of term occurences. We thus derive both the probability of a term occurring in a document ($p(t|d)$), and the probability of a term occurring in a collection of documents ($p(t|c)$). While the former directly corresponds to the *tf*, the latter allows a derivation of *idf*, via an inverse *log*, i.e.:

$$\text{idf} := -\log(p(t|c)) \quad (1)$$

To formulate the construction of said probabilities, our system allows for extensions of classical SQL, Datalog and relational algebra with probabilities for facts and rules, and additional operators for deriving probabilities from given relations. To derive the above term-frequency, a "Bayes" operator is applied to the *term*-relation, using the *context*-attribute as evidence:

```
CREATE VIEW tf AS
SELECT term, context
FROM term
EVIDENCE KEY 2;
```

This could equivalently be formulated in probabilistic relational algebra (PRA) [4] as:

```
tf = Bayes[$2](term);
```

where *Bayes* is a new operator introduced to the algebra. Yet another way of modelling this probability is by defining the *tf*-relation as a probabilistic relation, conditional on a context, in probabilistic Datalog [8] as:

```
tf(T,D)   :-   term(T,D) | (D).
```

Similarly, the *document*-probabilities can be defined, resulting in collection frequencies equivalent to *idf* in "classical" IR. To derive these probabilities in SQL, a view *idf* could be defined as:

```
CREATE VIEW idf AS
SELECT term
FROM tf
ASSUMPTION MAX_IDF;
```

where MAX_IDF is a probabilistic assumption that triggers the computation of term probabilities as defined for *idf*, as the log of the total number of documents divided by the number of documents containing a specific term.

3.3 Retrieval

Retrieval of indivdiual documents in this framework is carried out by joining the idf-weighted query terms with tf-weighted terms occurring in the document collection. This is expressed in SQL simply as

```
SELECT DISTINCT context
FROM tf, wqterm
WHERE qterm.term = tf.term;
```

where wqterm is the weighted variant of the (unweighted) query terms. This returns a list of (distinct) documents in which at least one of the query terms occurs. Due to the weighting of query- and document-terms, all documents returned have associated with them a weight that can be interpreted as a document's retrieval status value.

The same query could similarly be expressed in probabilistic Datalog as

```
retrieve(D)      :-   wqterm(T) & tf(T,D);
?-retrieve(D);
```

or in PRA as

```
?-PROJECT DISTINCT[$3]
     (JOIN[$1=$1](wqterm, tf));
```

The above examples only return document relevant to a given query, but for the ETREC, the goal was to find experts relevant to a given topic. It is therefore necessary to join the list of returned documents with a list of authors of documents, resulting in a list of authors (or experts) relevant to a given query. The implementation of this link is straightforward and follows the examples given above, however, we believe that the association between documents and authors should not be limited to a simple "one author per document" link, but should also be probabilistic, allowing for varying degrees of authorship.

3.4 Document-Expert association

When treating documents as sources of evidence for the expertise of expert-candidates, it is necessary to define an association between these sources of evidence and the candidates. The direct way of associating a document with an expert would be to model these links on the "sender"-information contained in the *meta*-tags of emails in the list collection. This approach, however, has some severe shortcomings. The ETREC collection does not only consist of emails, but also of other document-types, such as personal web-pages and CVS-repositories, for which information about the author might not be available. Furthermore, a simple link between documents and expert-candidates based on the sender information would not be able to capture expert-document links for documents authored by multiple persons, or capture links for documents which are *about* an expert-candidate, who is not the author of that document.

When QMIR participated in last year's ETREC, results for the expert finding task were rather low, compared to the results achieved by some of the other participants. It was assumed that this low performance was caused by an imprecision in the connection of documents and authors (expert candidates), as the "sender"-information alone was not sufficient to capture concepts such as multiple authorship and documents *about* a person.

To overcome these limitations, we chose to model the association between documents and expert-candidates as the probability of a document implying a candidate. A candidate can thus be seen as a query, reducing the implication to $p(d \rightarrow q)$ [10]. The queries were formed from the information on expert-candidates available as part of the ETREC-collection, with queries consisting of the names of experts and their e-mail addresses as query-terms.

To further gauge the impact different term-weighting strategies might have on modelling the association between expert-candidates and supportive documents, we modelled the association both with *tf-idf* and *Language Modelling* weighting.

4. EXPERIMENTS

Here, we describe our strategies for retrieving experts for given topics (queries) from the knowledge base. Queries are represented as described above. For our experiments, we concentrated on variations in the knowledge base used for expert finding, and the impact different knowledge bases might have on retrieval quality.

All the below strategies make use of some common relations for data representation, so they will be presented here, rather than in the individual subsections. As noted above, we used a probabilistic connection between documents and authors. This is represented in the system by an "author"-relation, that has two attributes, "candidate_id" to represent authors, and "context" to represent documents. Each tuple occurring in the relation is preceded by a weight, giving the strength of the association. On a physical level, there exist two different versions of this relation, one for the "lists" collection, and one for the "www" collection, however, conceptually, they are identical and will be treated as such for the present discussion.

Query terms are contained in a "qterm"-relation; joining this relation with the collection's idf values results in a relation called "wqterm", in which the query terms are weighted by the collection frequencies.

All other relations necessary for implementing the strategies will be explained in the respective sections.

4.1 WWW subcollection

In a first series of experiments, we limited the knowledge base to those documents which form the "www"-subcollection. This collection consists of personal webpages of W3C researchers, webpages of working groups etc. As mentioned in a previous section, there is no "given" author for any of the documents contained in this part of the collection, we therefore modelled authorship of documents in the collection based on the occurrence of expert-candidates' names or e-mail addresses. Retrieval of expert-candidates on the "www"-collection can be expressed in SQL as:

```
SELECT DISTINCT candidate_id, context
FROM wqterm, tf_www, www_author
WHERE wqterm.term = tf_www.term
AND tf_www.context = www_author.context;
```

An expression equivalent to the above SQL in PRA would be:

```
?-PROJECT DISTINCT[$4,$5](JOIN[$3=$2]
   (JOIN[$1=$1](wqterm,tf_www),
      www_author));
```

Similarly, the same strategy expressed in Datalog is:

```
retrieve(C,D)    :-  wqterm(T) &
                     tf_www(T,D) &
                     www_author(C,D);
?-retrieve(C,D);
```

Evaluating the performance of the individual strategies using trec_eval, the overall MAP-score achieved by the Body-only approach is 0.0965.

4.2 Lists subcollection

For the remaining experiments, only the "lists"-subcollection was used. As the "lists"-subcollection consists of emails sent to a mailing-list, we concentrated on variations in document structure specific in emails; specifically, we used quotations in emails as an additional source of evidence. The rationale behind different treatment of different email parts is that emails will usually only quote significant parts of previous messages. It could thus be argued that quotations in emails are a distilled version of previous emails, containing the salient parts.

4.2.1 Body-only

In the "body-only" approach, only the non-quoted parts of emails were used, while terms occurring in quotations inside emails were not considered for the expert finding process. The rationale behind this strategy is that quotations contain information not originally conceived by the author of the present email. Instead, the author quotes material from another email, which should be indicative of expertise of that other email's author. To exclude quotations from emails thus leads to a more "direct" association between terms and authors.

So, assuming that the "body"-only portion of emails is contained in a relation called "tf_body", which has two attributes - term and context - , the retrieval strategy is expressed in SQL as:

```
SELECT DISTINCT candidate_id, context
FROM wqterm, tf_body, lists_author
WHERE wqterm.term = tf_body.term
AND tf_body.context = lists_author.context;
```

This can equivalently be expressed in PRA as:

```
?-PROJECT DISTINCT[$4,$5](JOIN[$3=$2]
   (JOIN[$1=$1](wqterm,tf_body),lists_author);
```

And equivalently, in probabilistic Datalog as:

```
retrieve(C,D)    :-  wqterm(T) &
                     tf_body(T,D) &
                     lists_author(C,D);
?-retrieve(C,D);
```

Evaluating the performance of the individual strategies using trec_eval, the overall MAP-score achieved by the Body-only approach is 0.1376.

4.2.2 Quotes-only

The "quotes-only" approach is the exact opposite of the "body-only" approach, and only considers terms which occur inside quoted parts of emails. This follows the idea that quotations contain the most salient information in a thread of emails, and are therefore most indicative of expertise of authors participating in the discussion. The above argument that quotations contain information not originally conceived by the author of the present email still holds in this context, however, it could reasonably be argued that an author in a thread of emails related to a certain topic is still an expert on that topic, even if the most salient terms only occur in quotations. To make this more concrete, it can be reasonably argued that quotations in emails usually occur in replies to previous questions, or in general discussions regarding a specific topic. In either of these cases, an author of such an email (i.e. one containing quotations) might be an expert on the topics at hand.

The "quotes-only" strategy can be implemented just like the "body-only" strategy, the only difference being that instead of a "tf_body"-relation, a "tf_quotes"-relation is used to represent terms occurring in quotations.

Again using trec_eval for measuring the performance of the "quotes-only" approach, the overall MAP score achieved was 0.1242.

4.2.3 Body and Quotes

The "body+quotes" approach does not distinguish between terms forming quotations, and terms forming email bodies; it is thus most similar to standard document retrieval. This can be seen as a zero hypothesis to the above claim that utilising the structure of documents is beneficial to the task of expert finding.

Again, implementing this strategy is a straightforward extension of the previous examples; instead of a "tf_quotes" or "tf_body" relation, a "tf_bodyquotes"-relation is used, which represents terms present in either email bodies or quotations.

To investigate the influence term-weighting might have on the quality of retrieval, we modelled the retrieval

task for the *Body and Quotes* subcollection using both *tf-idf* and *Language Modelling*. Furthermore, we compared the quality of retrieval using a stemmed and an unstemmed version of the subcollection, to prevent query-mismatch problems due to overstemming. In conjunction with the two strategies for building document-expert associations (using *tf-idf* and *Language Modelling*), we thus had a total of eight runs: stemmed *tf-idf* using *tf-idf* association, stemmed *tf-idf* using *LM* association, unstemmed *tf-idf* using *tf-idf* association, unstemmed *tf-idf* using *LM* association, stemmed *LM* using *tf-idf* association, stemmed *LM* using *LM* association, unstemmed *LM* using *tf-idf* association and unstemmed *LM* using *LM* association.

The overall MAP score for the *Body and Quotes* approach is 0.1108. Looking at the performance of individual queries, we find that some of the queries fail to return any relevant experts for a query, while other queries find relevant experts with a mean average precision as high as 0.5. The results for the experiments using a stemmed representation of the collection can be seen in Figure 1.

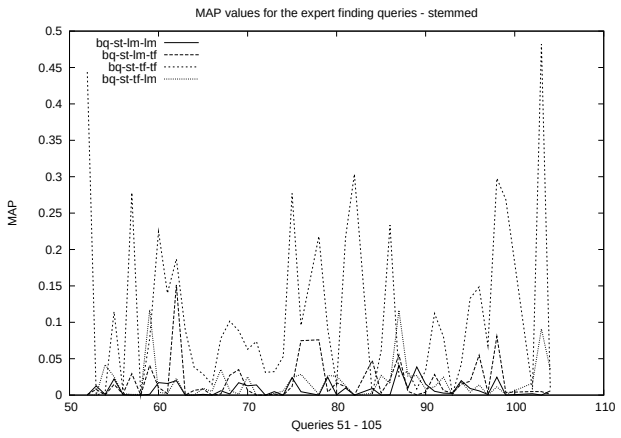

Figure 1: Map values for stemmed bodyquotes

The MAP scores show that the best retrieval quality is achievable using a combination of *tf* collection term weighting and *tf*-weighted candidate association.

The results for experiments with an unstemmed representation of the collection, as shown in Figure 2 show similar results. Here, applying *tf*-weighting for both document retrieval and candidate association works best, however, for some of the queries, a combination of *LM*-weighting on term and *tf* on candidate associations performs similarly well as a *tfidf-tfidf* combination, or even outperforms it.

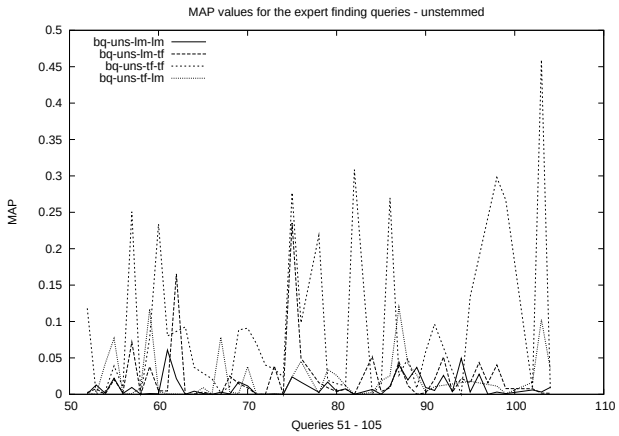

Figure 2: MAP values for unstemmed bodyquotes

Our approach delivers the relevant candidates for about 20% of the queries, with MAP scores as high as 0.48. The results also display low MAP scores for some of the queries, significantly lowering the overall MAP scores for the different approaches. Furthermore, the results confirm the findings of the 14th TREC E-TREC overview ([1]) that high precision retrieval of experts requires the combination of multiple sources of evidence.

	bq-lm-lm	bq-lm-tf	bq-tf-tf	bq-tf-lm
high	0.0408	0.1513	0.4822	0.1172
avg	0.0082	0.0184	0.1108	0.0179
#Q > avg	19	14	17	18

Table 1: MAP statistic for stemmed bodyquotes

The previously mentioned advantage of the *tfidf-tfidf* approach is also visible in Table 1. The highest MAP achieved by any query with this approach is 0.4822. The number of queries which perform better than the average MAP of 0.1108 is 17, again indicating that the overall performance is affected by a large number of queries that do not perform well. Also note that for the other approaches, the number of queries which perform better than the average is similar, however, due to the low average this is probably not significant.

The results for the experiments with a stemmed version of the bodyquotes collection were largely mirrored by the experiments with the unstemmed counterpart (Table 2). Here, too, the *tfidf-tfidf* approach performs best, with a highest MAP of 0.4594. However, the *LM*-textittfidf approach works better on the unstemmed representation than on the stemmed one.

	bq-lm-lm	bq-lm-tf	bq-tf-tf	bq-tf-lm
high	0.0608	0.2353	0.4594	0.1195
avg	0.0103	0.0221	0.0908	0.0223
#Q > avg	16	12	14	15

Table 2: MAP statistics for unstemmed bodyquotes

4.3 Results

The MAP-scores for the individual strategies indicate that the "body-only" approach performs best in identifying the correct experts for a given query with a MAP of 0.1376. This is followed by the "quotes-only" approach with a MAP score of 0.1242. The "body-quotes" approach, which does not distinguish between different elements of emails only achieves an overall MAP of 0.1108. The "www-only" approach performs worst with a MAP of 0.0965.

	www	body-only	quotes-only	body-quotes
MAP	0.0965	0.1376	0.1242	0.1108

Table 3: MAP values for the different approaches

More interesting are the results achieved for the different term-weighting and document-candidate association models. Results here show that a stemmed representation clearly outperforms an unstemmed representation, and, more surprisingly, that approaches based on pure *tf-idf* perform better than pure *LM* approaches or mixtures of *tf-idf* and *LM*.

5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

The Enterprise TREC provides an interesting framework in which to experiment with the integration of information retrieval and DBMS techniques: the ETREC data is fairly large and heterogeneous, as is the case with data typically encountered in commercial organisations, but requires the application of typical information retrieval tasks. Our experiments show that our probabilistic IR framework is able to cope with the amount of information provided, and that it is possible to express strategies for expert finding in a simple, straightforward way, without making any changes to the underlying system.

5.2 Future Work

The approaches presented in this paper all implement retrieval strategies at a fairly low abstraction level. For expert finding tasks to be applicable by non-expert users, it would need to be possible to express the strategies at a higher abstraction level. Furthermore, expert finding can be seen as a two-phase operation: first, finding documents relevant to a query, followed by associating those documents with expert candidates. Another way of finding experts given a query would be to first aggregate all documents associated with an expert candidate, and then summarise those documents, to generate a profile for that candidate. Such a profile could then be used as a surrogate for the full document collection when querying for an expert.

Both of the above issues could be addressed by a *summarisation logic*. A summarisation logic takes away implementational details, and provides users with a means to express different summarisation strategies (e.g. using stemmed or unstemmed document representations, or using different probabilistic models) via a more abstract syntax, coupled with a well-defined semantics. For this, we propose our summarisation logic POLIS, which uses structural information of knowledge representations to derive a summary. Further research will be necessary to evaluate the quality of expert profiles derived in this fashion.

6. REFERENCES

[1] Nick Craswell, Arjen P. de Vries, and Ian Soboroff. Overview of the trec-2005 enterprise track.

[2] Nick Craswell, David Hawking, Anne-Marie Vercoustre, and Peter Wilkins. Panoptic expert: Searching for experts not just for documents. In *Ausweb Poster Proceedings*, Queensland, Australia, 2001.

[3] Leonard N. Foner. Yenta: a multi-agent, referral-based matchmaking system. In *AGENTS '97: Proceedings of the first international conference on Autonomous agents*, pages 301–307, New York, NY, USA, 1997. ACM Press.

[4] N. Fuhr and T. Rölleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Transactions on Information Systems*, 14(1):32–66, 1997.

[5] David Hawking. Challenges in enterprise search. In *ADC*, pages 15–24, 2004.

[6] Tim Reichling, Andreas Becks, Oliver Bresser, and Volker Wulf. Kontaktanbahnung in Lernplattformen. In *Mensch & Computer 2004: Allgegenwärtige Interaktion*, pages 179–188. Oldenbourg Verlag, 2004.

[7] T. Roelleke. *POOL: Probabilistic Object-Oriented Logical Representation and Retrieval of Complex Objects*. Shaker Verlag, Aachen, 1999. Dissertation.

[8] Thomas Rölleke and Norbert Fuhr. Information

retrieval with probabilistic Datalog. In *Logic and Uncertainty in Information Retrieval: Advanced models for the representation and retrieval of information*, chapter 9, pages 221–245. Kluwer Academic Publishers, Boston et al., 1998.

[9] Thomas Rölleke, Ralf Lübeck, and Gabriella Kazai. The HySpirit retrieval platform. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, page 454, New York, NY, USA, 2001. ACM Press.

[10] C. J. van Rijsbergen. A non-classical logic for information retrieval. *Comput. J.*, 29(6):481–485, 1986.

[11] D. Yimam-Seid and A. Kobsa. Demoir: A hybrid architecture for expertise modeling and recommender systems.