UMass at TREC 2006: Enterprise Track

Desislava Petkova and W. Bruce Croft
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts, Amherst, MA 01003

Abstract

This paper gives an overview of the work done at the University of Massachusetts, Amherst for the TREC 2006 Enterprise track. For the discussion search task, we compare two methods for incorporating thread evidence into the language models of email messages. For the expert finding task, we create implicit expert representations as mixtures of language models from associated documents.

1 Introduction

In this paper we discuss two retrieval tasks in an enterprise setting: finding email messages which contain pro or con arguments on a given topic, and finding people who are knowledgeable in particular area. Both tasks offer unique challenges that are particular to enterprise search: diversity of structured and unstructured documents, variety of languages and formats, existence of social networks, security requirements. In our work we consider some of these issues, in particular document structure and heterogeneity.

2 Data processing

For our experiments, we used the Indri search engine in the Lemur toolkit [5]. Its powerful query language allows formulating richly structured queries and incorporating various sources of contextual evidence.

We preprocessed the *www* and *lists* subcollections of the W3c corpus, removing HTML tags from web documents and stripping quoted text and signature lines from email documents. We used Jangada [1] to extract signature blocks and reply-to lines from emails. Finally, we used the *all-in-reply-to* list compiled by William Webber to group emails by thread.

3 Discussion Search

Emails form a considerable part of the communication in an organization and are characterized by rich internal and external structure. Previous work has shown that email structure is a useful source of information in known-item finding [4].

We exploit internal structure by weighting header and mainbody text differently and external structure by adding a third component corresponding to thread text. In the header we combine evidence from the subject, date, to, from and cc fields. The mainbody is the original text of a message with reply-to, forward and signature lines removed. And the thread is the concatenated text of messages in the tree-like structure of an email conversation.

One approach for incorporating thread context is to estimate language model for the thread and interpolate it with the smoothed language models of the other email components. This corresponds to the following retrieval model:

$$\begin{aligned} P(t|D) &= \lambda_1 P(t|D_{header}) + \lambda_2 P(t|D_{mainbody}) \\ &+ \lambda_3 P(t|D_{thread}) \end{aligned}$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and

$$\begin{aligned} P(t|D_{component}) &= P_{MLE}(t|D_{component}) \\ &+ \alpha P(t|Corpus) \end{aligned}$$

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE
UMass at TREC 2006: Enterprise Track

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Massachusetts,Center for Intelligent Information Retrieval,Department of Computer Science,Amherst,MA,01003

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**4**	

with α being a general symbol for smoothing.

An alternative way to take advantage of thread information is to use it as a background model for smoothing maximum likelihood estimates from the header and mainbody. The hypothesis is that threads will provide a more reasonable fallback distribution than a word distribution for general English.

$$\begin{aligned} P(t|D) &= \lambda_1 P(t|D_{header}) \\ &+ \lambda_2 P(t|D_{body}) \end{aligned}$$

where

$$\begin{aligned} P(t|D_{component}) &= P_{MLE}(t|D_{component}) \\ &+ \alpha P(t|D_{thread}) \\ P(t|Thread) &= P_{MLE}(t|Thread) \\ &+ \beta P(t|Corpus) \end{aligned}$$

3.1 Runs

We submitted four official runs for evaluation. For all runs we the Term Dependency Model [3] and the Relevance Model [2] for pseudo-relevance feedback to expand the query. Term dependency increases precision by adding proximity features and pseudo-relevance feedback increases recall by adding terms related to the original query. We use the following mixing weights for the components models: $\lambda_{header} = 0.3$,$\lambda_{mainbody} = 0.7$,$\lambda_{thread} = 0.3$.

1. `Baseline`: (Unofficial) Uses the entire email content (header fields and mainbody) without breaking it up into components.

2. `UMaTiMixHdr`: Emails are divided into header and mainbody. The components are smoothed independently with background models of all header text and all mainbody text.

3. `UMaTiSmoThr`: The header and mainbody components are both smoothed with the thread which a particular message is part of. (Messages are not part of a thread are smoothed with the collection, as in `UMaTiMixHdr`.)

4. `UMaTiMixThr`: Emails have three components - header, mainbody and text, smoothed independently with background models of header, mainbody and thread text.

5. `UMaTDMixThr`: Similar to `UMaTiMixThr` but also using the description field of the query topic.

Run ID	mAP	R-prec	RR1	P@5	P@10
`Baseline`	.2905	.3367	.6834	.5240	.4700
`UMaTiMixHdr`	.3058	.3567	.6673	.5520	.5240
`UMaTiSmoThr`	.3197	.3672	.7081	.5800	.5340
`UMaTiMixThr`	.3373	.3715	.6538	.5440	.5300
`UMaTDMixThr`	.3631	.3963	.7134	.5880	.5820

Table 1: Discussion search: submitted runs.

Results show that smoothing with a thread-based fallback model is more effective than smoothing with a general collection model. However, constructing a mixture of language models from header, main body and thread text is even more effective. And corroborating previous research, internal and external email structure is an useful source of information.

4 Expert Search

We propose a formal method for constructing hierarchical expert representations, based on statistical relevance modeling. Our main goal is that this process takes advantage of various information sources and prior knowledge about the experts, the collection or the domain.

4.1 Model

Our expert modeling approach includes the following steps:

1. Define what is a reference to E, so that occurrences of E can be detected.

2. Rank and retrieve profile documents. We use language modeling with Dirichlet smoothing.

3. For each document D in the profile set S_E, compute the posterior $P(D|E)$, assuming that the prior distribution is uniform.

$$P(D|E) = \frac{P(E|D)P(D)}{P(E)}$$

4. Form a term distribution for E by incorporating the document model $P(t|D)$ and marginalizing. $P(t|D)$ is the maximum likelihood estimate.

$$P(t|E) = \sum_{D \in S_E} P(t|D)P(D|E)$$

To summarize, we represent an expert as a mixture of documents, where the mixing weights are specified by the posterior distribution $P(D|E)$. Once we have built models for all candidates, we find experts relevant to a particular topic Q by ranking the candidates according to query likelihood.

The result of Eq. (4) is a probability distribution of words describing the context of an expert's name, where $P(\cdot|E)$ is estimated using a particular name definition and from a homogeneous collection of documents. Representations estimated from different collections or alternative name definitions can be interpolated to build hierarchical expert models.

$$P(t|E) = \sum_{c \in \mathcal{C}} \lambda_c P(t|E_c), \sum_{c \in \mathcal{C}} \lambda_c \;=\; 1$$

4.2 Experiments

With our experiments we want to determine whether hierarchical expert models can effectively create rich representations from heterogeneous data and answer complex queries.

Heterogeneous sources

The W3C corpus is composed of several subcollections comprising documents of particular type. We independently build a language model from one subcollection at a time and then represent an expert as a mixture of those models. This allows us to treat each subcollection differently according to its specific intrinsic properties, e.g. when smoothing to estimate $P(E|D)$, as well as to weight the information sources.

We use the *lists* subcollection (average length 450 words after preprocessing) and the *www* collection (average length 2000). We automatically set the Dirichlet smoothing μ parameter to the average document length, and we experimentally determine an optimal value for the mixing parameter $\lambda_{www} = 0.6$. Results are reported in Table 2. Although models built from *www* outperform models built from *lists*, by combining the two we achieve an even better performance, indicating that email discussion lists contain information not contained in the web pages.

Term dependency

An important feature of our model is that it preserves information inherent in documents, such as structure and term positions. Consequently, it can capture higher-level language features such as relationships between terms. Note that because experts are modeled indirectly as a set of documents, it is possible that query terms appear in the profile set but do not co-occur in any document within that set.

We implement term dependency as described by Metzler and Croft [3], using both sequential dependency and full dependency between query terms to include restrictions on terms appearing in close proximity in the text. Results (Table 2) show consistent improvement in mean average precision.

Combining expert definitions

Finally we compare two rules to define a person entity: *LAST* matches the last name of a candidate, and *FULL* matches the first and last name separated by at most two words. *LAST* is a loose definition since some people have the same family name. *FULL* is more strict but misses some true associations, since people are not necessarily referred to with their full names, especially in emails.

This is an example of the tradeoff between recall and precision. The profile set from a loose definition is larger but more ambiguous as some associations are incorrect. On the other hand, the profile set from a strict definition is smaller but more precise as retrieved documents are reliably associated with the person but at the same time valid documents are overlooked. Combining two expert definitions, *LAST* and *FULL* gives a slightly better performance than either alternative separately (Table 2).

Entity definition		Documents		Term dependency	
LAST	*FULL*	*lists*	*www*	No	Yes
X		X		.1841	.2008
X			X	.2387	.2922
	X	X		.2679	.3029
	X		X	.3127	.3732
X		X	X	.2689	.3220
	X	X	X	.3460	.4139
X	X	X	X	.3514	.4216

Table 2: mAP is incrementally improved by combining several representations to form a hierarchy. Mixing parameters: $\lambda_{FULL} = 0.9$,$\lambda_{LAST} = 0.1$,$\lambda_{www} = 0.6$,$\lambda_{lists} = 0.4$

4.3 Runs

We submitted four official runs which use the hierarchical representation described above. We expand the title-only query with the description and narrative fields, and with pseudo-relevance feedback terms.

1. `UMaTiDm`: Title / Term dependency
2. `UMaTNDm`: Title+Description / Term dependency
3. `UMaTNFb`: Title+Description / Term dependency / Pseudo relevance feedback
4. `UMaTDFb`: Title+Description+Narrative / Term dependency / Pseudo relevance feedback

Run ID	mAP	R-prec	RR1	P@5	P@10
`UMaTiDm`	.4216	.4379	.7478	.6449	.5469
`UMaTNDm`	.4307	.4573	.7875	.6408	.5469
`UMaTNFb`	.4706	.4972	.8207	.6980	.5816
`UMaTDFb`	.5016	.5108	.8571	.7265	.6388

Table 3: Expert finding: submitted runs.

4.4 Conclusion

We described a language modeling approach for finding people who are experts on a given topic. It is based on collecting evidence for expertise from multiple sources in a heterogeneous collection, using language modeling to find associations between documents and experts and estimate the degree of association, and finally integrating models to construct rich and effective expert representations.

Our approach provides a broad framework for answering questions about experts. It is based on the following two assumptions: 1. We can come up with useful definition(s) of a named entity in the form of a query. This allows us to use any retrieval technique to find associated documents. 2. We can assume that the co-occurrence of terms and named entities is evidence of topical connection. Note that this means we do not distinguish between positive and negative document support. On the other hand, since the model does not have any specific knowledge about what it means to be an expert, it is very general and can be applied to building representations for other named entities such as places, events, organizations.

5 Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under contract number NBCHD030010. Any opinions, findings and conclusions expressed in this material are the author's and do not necessarily reflect those of the sponsor.

References

[1] V. R. Carvalho and W. W. Cohen. Learning to extract signature and reply lines from email. In *CEAS '04*.

[2] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR '01*.

[3] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *SIGIR '05*.

[4] P. Ogilvie and J. Callan. Experiments with language models for known-item finding of e-mail messages. In *TREC '05*.

[5] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. Technical report, UMass, Amherst, 2005.