Experiments with Language Models for Known-Item Finding of E-mail Messages

Paul Ogilvie and Jamie Callan
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
pto@lti.cs.cmu.edu, callan@lti.cs.cmu.edu

Abstract

We present experiments using language models to rank e-mail messages for the Known-Item Finding task of the Enterprise track. We combine evidence from the text of the message, its subject, the text of the thread the in which the message occurs, and the text of messages that are in reply to the message. We find that the only statistically significant differences suggest that in addition to the text of the message, the subject is a very important piece of evidence. We also explore the use of a depth based prior, where emphasis is place on messages near the root of the thread structure, which has mixed results.

1 Introduction

In this paper we consider the task of known-item finding of email messages equivalent to other structured retrieval tasks. In doing so, we observe that email messages occur in a thread structure as well as have their own internal structure. Viewed in this way, we can consider this task to be identical to the XML component retrieval investigated at INEX [2]. We consider "documents" to be a thread of email messages and the components we wish to retrieve as the email messages, which are components contained in the documents.

Structuring email messages in this manner allows us to easily investigate incorporating evidence from the thread as well as structure within the email message itself. We do so using code to be soon released (scheduled for January 2007) as part of the Indri search engine in the Lemur toolkit [1].

We use the Indri search engine to investigate two methods to combine evidence. One is a hierarchical language modeling approach investigated at INEX [4][6] and the other is a post query combination method investigated for known-item finding of web pages [5]. Within these methods we investigate the use of the text of the emails, the email subjects, the text of the thread, and that of the subthread (messages in reply to a message). We also explore the use of a

depth based prior, where depth corresponds to the depth of the message in the thread. The depth based prior was motivated by observations of the training topics.

Section 2 presents the mixture method followed by the post query combination method. It also describes how the depth prior is incorporated into ranking. Section 3 describes the implementation in Indri, estimation of the prior, and the evaluation methodology. Section 4 presents results on the training topics, while Section 5 presents the results on the test topics. The paper is concluded in Section 6.

2 Model

In one approach, we rank email messages by estimating the probability that the a language model estimated for the email message generated the query. We use simple unigram language models, which are multinomial probability distributions over words in the vocabulary. That is, a language model θ specifies $P(w|\theta)$. Email messages are then ordered by $P(Q|\theta_e) = \prod_{i=1}^{|Q|} P(q_i|\theta_e)$ where θ_e is the language model estimated for a particular email message e.

In order to estimate the language model θ_e, we note that we would like to incorporate evidence from the message itself and from other messages in the thread in which e was found. With that in mind, we estimate θ_e as a linear combination of several language models:

$$\begin{aligned} P(w|\theta_e) = \quad & \lambda_{em} P\left(w \middle| \theta_{MLE(em)}\right) \\ & + \lambda_{ti} P\left(w \middle| \theta_{MLE(ti)}\right) \\ & + \lambda_{th} P\left(w \middle| \theta_{MLE(th)}\right) \\ & + \lambda_{su} P\left(w \middle| \theta_{MLE(su)}\right) \\ & + \lambda_{co} P\left(w \middle| \theta_{MLE(co)}\right) \end{aligned} \tag{1}$$

where *em* refers to the text of the email, *ti* refers to the title tag in the documents (the subject of the email), *th* refers to the text of the entire thread, *su* refers to the text of the subthread which corresponds to the email itself and all replies to the email, and *co* refers to the text of the entire collection. $\theta_{MLE(text)}$ is the Maximum Likelihood Estimate of the multinomial language model on *text*, which is given by:

$$P\left(w \middle| \theta_{MLE(text)}\right) = \frac{\text{count of } w \text{ in } text}{\text{length in words of } text} \tag{2}$$

The other approach we investigate performs an equivalent of a meta-search ranking model where the different rankings are produced by the different ranking models. Here we rank by

$$P(Q|\theta'_{em})^{\lambda_{em}} P(Q|\theta'_{ti})^{\lambda_{ti}} P(Q|\theta'_{th})^{\lambda_{th}} P(Q|\theta'_{su})^{\lambda_{su}} \tag{3}$$

where

$$P(w|\theta'_x) = (1-\lambda_{co}) P\left(w \middle| \theta_{MLE(x)}\right) + \lambda_{co} P\left(w \middle| \theta_{MLE(co)}\right) \tag{4}$$

This model does allow relative weighting of the different structural components of messages in the thread. This model corresponds to the linear weighted combination of log probabilities, which we investigated in [5]. We will present results on ranking by $P(Q|\theta_e)$ and by the model specified in Equation 3. We will refer to ranking by $P(Q|\theta_e)$ as the mixture method and Equation 3 as the post query combination approach. Our official submissions used the post query combination approach

Inspired by the usefulness of document priors for the similar task of known-item finding on the Web [3], we experimented with the use of prior probabilities of relevance based on the "depth" of the message. By "depth", we mean refer to the depth of the email in its thread structure. To incorporate the depth prior into either model, we multiple the score ($P(Q|\theta_e)$ or Equation 3) by:

$$P(e \text{ is relevant} | depth(e)) \tag{5}$$

which we estimate from the training data.

3 Methodology

This section briefly discusses technical details of the implementation and the evaluation techniques that will be used in later sections.

3.1 Implementation

Our experiments were performed in a locally enhanced version of the Indri search engine, which is a part of the Lemur toolkit [1]. The corpus was reorganized so that all messages in a thread are in the same document, where an email message was restructured as:

```
<lists>
[trec formatted document]
  <responses>
    [response 1]
    [response 2]
    ...
    [response n]
  </responses>
</lists>
```

where "[trec formatted document]" refers to an email in the w3c corpus with its "<DOC>" and "</DOC>" tags and a "response" is a reply to the message formatted as other email messages surrounded by "<lists>" and "</lists>" tags and also containing its responses. Thread structure was constructed from William Webber's in-reply-to file [9].

Within Indri, we indexed the "lists", "responses", "DOC", and "title" tags, where the "title" tag corresponds to the subject of the original email. We

$depth$	Posterior $P(depth(E)\|E \text{ is rel.})$	$P(depth(E))$	$\propto$ Prior $\propto P(E \text{ is rel.}\|depth(E))$
$= 0$	0.893	0.674	1.32
$= 1$	0.071	0.171	0.42
≥ 2	0.036	0.154	0.23

Table 1: Estimation of the thread depth based prior.

stemmed terms using the Krovetz stemmer, but we did not use a stopword list. The original topics were converted into NEXI [7] queries to search the structured database using the simple rewrite:

```
//doc[about(., [topic terms])]
```

where "[topic terms]" are the terms contained in the original topic. This query simply requests "doc" components (email messages) matching the original query.

3.2 Prior Estimation

We estimated the email message depth based prior for three categories of depths: zero, one, and two or more. To estimate the prior probability in Equation 5, we first estimate the posterior $P\left(depth\left(E\right)|E \text{ is relevant}\right)$. We can estimate this directly from the training topics. When performing estimation, we used a Laplace estimator of the posterior in the topic set, as we had only 25 training topics:

$$\begin{aligned} P\left(depth\left(E\right)=0\,|E \text{ is relevant}\right) &= \frac{count(depth=0)+1}{number\ topics+3} = \frac{25}{28} = 0.893 \\ P\left(depth\left(E\right)=1\,|E \text{ is relevant}\right) &= \frac{count(depth=1)+1}{number\ topics+3} = \frac{2}{28} = 0.071 \\ P\left(depth\left(E\right)\geq 2\,|E \text{ is relevant}\right) &= \frac{count(depth\geq 2)+1}{number\ topics+3} = \frac{1}{28} = 0.036 \end{aligned} \tag{6}$$

The posterior distribution was used to estimate the prior probability of relevance for an email through the use of Bayes rule:

$$P\left(E \text{ is relevant}\,|depth\left(E\right)\right) = \frac{P\left(depth\left(E\right)|E \text{ is relevant}\right)P\left(E \text{ is relevant}\right)}{P\left(depth\left(E\right)\right)} \tag{7}$$

$P\left(E \text{ is relevant}\right)$ was assumed constant across all email messages and $P\left(depth\left(E\right)\right)$ was estimated by examination of email messages in the corpus. Note that we only estimated a value proportional to the prior during ranking, as $P\left(E \text{ is relevant}\right)$ was assumed constant. The values we estimated are presented in Table 1.

3.3 Measures

For evaluation measures we present the number found in the top 100 and mean reciprocal rank (MRR), placing emphasis on MRR. Mean reciprocal rank is the

average over all queries of one divided by the rank the correct document was found in the top 100 results (zero if there is no correct document in the top 100 results). A MRR close to one is good and MRR places much emphasis on the systems ability to rank correct documents near the top of the list.

When we discuss statistical significance tests, we use a one-tailed pairwise comparison using the bootstrap and the Benjamini-Hochberg method for multiple test correction [8]. The bootstrap uses repeated samples of topics with replacement from the original topic set. MRR is computed over the samples. This allows the bootstrap to non-parametrically estimate what may happen on other topic sets. From this we can estimate whether a system configuration performs significantly better than another configuration of the system. As we do these comparisons many times, it is important to correct for multiple testing. Otherwise, we may incorrectly conclude that two system configurations behaved differently when the differences could be easily due to random chance. To correct for this, we use the Benjamini-Hochberg method. For more information on the bootstrap and correction for multiple tests, see [8]. The significance tests reported below use 5000 bootstrap samples and significance is tested at the 0.05 level.

4 Training Topics

Tables 2 and 3 show the results of various combinations of using the subject, thread, and subthread information was well as the depth prior of the post query combination method and the mixture combination method. The parameter values were hand chosen and may not be optimal for either the training or testing set. From the tables we see that there is a trend that the subject and depth priors tend to help performance noticeably. We also observe that the post query combination method seems to perform better than the mixture method.

In order to get a better understanding of the differences found between configurations of the system in the training data, we performed one-tailed pairwise comparisons using the bootstrap test corrected for multiple testing using the Benjamini-Hochberg method. After running statistical significance tests, we found that there were very few significant differences at the 0.05 level.

The few cases where the significance test did find differences between systems involved comparing the post query combination method with weight on the subject of the email with the mixture language model method that did not place extra weight on words in the subject of the email, but did place some weight on the thread or subthread.

If we rely on the values for MRR on the training set, we would guess that the post query combination method using all features and the depth prior would result in the best performance on the evaluation set of topics. However, noting that this configuration was not significantly better than any of the other approaches tested, one should not be very confident in this guess.

Dirichlet Parameter μ	Subject λ_{ti}	Thread λ_{th}	Subthread λ_{su}	Depth Prior	Number Found	MRR
700	0.22	0.2	0.02	YES	22	0.567
700	0.22	0.2	0.02	NO	22	0.534
700	0.22	0.2	0	YES	22	0.568
700	0.22	0.2	0	NO	22	0.536
700	0.22	0	0.02	YES	22	0.568
700	0.22	0	0.02	NO	22	0.538
700	0.22	0	0	YES	22	0.568
700	0.22	0	0	NO	22	0.540
700	0	0.2	0.02	YES	22	0.507
700	0	0.2	0.02	NO	22	0.455
700	0	0.2	0	YES	22	0.507
700	0	0.2	0	NO	22	0.455
700	0	0	0.02	YES	22	0.501
700	0	0	0.02	NO	22	0.451
700	0	0	0	YES	22	0.506
700	0	0	0	NO	22	0.451

Table 2: Results of various configurations of the post query combination method (Equation 3) evaluated on the training topics.

Dirichlet Parameter μ	Subject λ_{ti}	Thread λ_{th}	Subthread λ_{su}	Depth Prior	Number Found	MRR
700	0.72	0.2	0.02	YES	22	0.526
700	0.72	0.2	0.02	NO	22	0.500
700	0.72	0.2	0	YES	22	0.546
700	0.72	0.2	0	NO	22	0.502
700	0.72	0	0.02	YES	22	0.550
700	0.72	0	0.02	NO	22	0.520
700	0.72	0	0	YES	22	0.550
700	0.72	0	0	NO	22	0.520
700	0	0.2	0.02	YES	22	0.506
700	0	0.2	0.02	NO	22	0.453
700	0	0.2	0	YES	22	0.506
700	0	0.2	0	NO	22	0.453
700	0	0	0.02	YES	22	0.501
700	0	0	0.02	NO	22	0.451
700	0	0	0	YES	22	0.506
700	0	0	0	NO	22	0.451

Table 3: Results of various configurations of the mixture method ($P(Q|\theta_e)$) evaluated on the training topics.

μ	Subj. λ_{ti}	Thr. λ_{th}	Subthr. λ_{su}	Depth Prior	Num. Found	MRR	Official MRR	Run Name
700	0.22	0.2	0.02	YES	108	0.589	0.582	CMUallon
700	0.22	0.2	0.02	NO	113	0.591	0.598	CMUnoprior
700	0.22	0.2	0	YES	108	0.588		
700	0.22	0.2	0	NO	113	0.591		
700	0.22	0	0.02	YES	108	0.589		
700	0.22	0	0.02	NO	113	0.591	0.601	CMUnoPS
700	0.22	0	0	YES	108	0.592		
700	0.22	0	0	NO	113	0.588	0.596	CMUnoPSD
700	0	0.2	0.02	YES	107	0.560		
700	0	0.2	0.02	NO	111	0.531		
700	0	0.2	0	YES	107	0.560		
700	0	0.2	0	NO	111	0.531		
700	0	0	0.02	YES	106	0.551		
700	0	0	0.02	NO	112	0.525		
700	0	0	0	YES	106	0.550		
700	0	0	0	NO	112	0.524	0.525	CMUnoPSDT

Table 4: Results of various configurations of the post query combination method (Equation 3) evaluated on the training topics. Our official runs have different results than those presented in this table. We have presented the performance of the official runs next to similar system configurations. We are still looking for the source of the disagreement in performance.

5 Test Topics

Tables 4 and 5 show performance of the post query combination and mixture methods on the test topics. Due to some system differences, our official submissions had higher MRR than the similar system configurations in Table 4. We have presented the official results beside these results for clarity.

We first note that the depth based prior did not always improve performance as it did in the training set. This suggests that our training set was not representative of the test set with regards the thread depth of a known-item messages.

Another thing to observe is that the subject of the email message consistently provides valuable information for both the post query combination method and the mixture method. The mixture method seems to get additional benefit from the thread and subthread, while the post query combination method does not.

We also wish to observe that the lower performance of the mixture method when compared to the post query combination method on the test set has disappeared in the test topics. In fact, of the unofficial results, the mixture method using all features except the depth prior has the best mean reciprocal rank.

However, most of these differences are not statistically significant. As with

Dirichlet Parameter μ	Subject λ_{ti}	Thread λ_{th}	Subthread λ_{su}	Depth Prior	Number Found	MRR
700	0.72	0.2	0.02	YES	109	0.581
700	0.72	0.2	0.02	NO	111	0.599
700	0.72	0.2	0	YES	106	0.552
700	0.72	0.2	0	NO	111	0.595
700	0.72	0	0.02	YES	105	0.573
700	0.72	0	0.02	NO	111	0.585
700	0.72	0	0	YES	105	0.582
700	0.72	0	0	NO	111	0.587
700	0	0.2	0.02	YES	106	0.552
700	0	0.2	0.02	NO	111	0.537
700	0	0.2	0	YES	106	0.551
700	0	0.2	0	NO	111	0.536
700	0	0	0.02	YES	106	0.550
700	0	0	0.02	NO	111	0.526
700	0	0	0	YES	106	0.550
700	0	0	0	NO	112	0.524

Table 5: Results of various configurations of the mixture method ($P(Q|\theta_e)$) evaluated on the training topics.

the test topics, we also found few significant differences between the system configurations in the test topics. Variations of the post query combination method placing weight on the subject but not using the depth prior were better than both mixture and post query combination methods that did not use the subject or the depth prior. These observations are fairly consistent with those found in the training set, although more significant differences were found.

6 Conclusions

We investigated the use of two combination methods to combine evidence in email messages for the task of known-item finding of email messages. The post query combination method is essentially a meta-search combination method that ranks each email document representation, then combines the scores from each representation using a weighted sum of the log of the scores from the representations. The mixture method combines the language models estimated from each representation. Both methods performed well on the corpus, although the mixture combination method may be more stable with the addition of new features.

When using the combination methods, we examined the subject of the messages as well as the content of the thread and subthread (all messages in reply to the message). The only universally helpful feature was the subject of the thread, which improved performance noticeably for both combination methods. In some cases, this improvement was statistically significant at the 0.05 level in

both the training and testing topics. The only statistically significant differences found between configurations of the systems involved configurations using the subject performing better than configurations that did not use the method.

We also considered a depth based prior that looked promising on the training topics but did not perform well on the test topics. We would like to investigate this more for the final version of this paper.

We would also like to note that this was in part an experiment in using the Lemur toolkit [1]. Apart from the depth prior (which proved to not be very effective) and some data conversion scripts, we were able to use the toolkit without modification. This demonstrates the flexibility and effectiveness of the Lemur toolkit.

7 Acknowledgements

The authors would like to thank William Webber for providing the extracted thread structure. This research was sponsored by National Science Foundation (NSF) grant no. CCR-0122581. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implicit, of the NSF or the US government.

References

[1] The lemur toolkit for language modeling and information retrieval. http://lemurproject.org/.

[2] N. Fuhr, S. Maalik, and M. Lalmas, editors. *Proc. of the Second Annual Workshop of the Initiative for the Evaluation of XML retrieval (INEX)*, Dagstuhl, Germany, Dec. 2003.

[3] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 27–34. ACM, 2002.

[4] P. Ogilvie and J. Callan. Language models and structured document retrieval. In *Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval (INEX)*, 2003.

[5] P. Ogilvie and J. P. Callan. Combining document representations for known-item search. In *Proc. of the 26th annual int. ACM SIGIR conf. on Research and development in informaion retrieval (SIGIR-03)*, pages 143–150, New York, July 28– Aug. –1 2003. ACM Press.

[6] P. Ogilvie and J. P. Callan. Using language models for flat text queries in xml retrieval. In *Proc. of the Second Annual Workshop of the Initiative for the Evaluation of XML retrieval (INEX)*, Dagstuhl, Germany, Dec. 2003.

[7] A. Trotman and B. Sigurbjörnsson. Narrow Extended XPath I. Technical report, 2004. Available at http://inex.is.informatik.uni-duisburg.de:2004/.

[8] L. Wasserman. *All of Statistics*. Springer, 2004.

[9] W. Webber. Thread structure of w3c lists. http://www.cs.mu.oz.au/ wew/w3c-lists-threads-wew.tar.gz.