Using Contextual Information to Improve Search in Email Archives

Wouter Weerkamp, Krisztian Balog, and Maarten de Rijke

ISLA, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
w.weerkamp@uva.nl, k.balog@uva.nl, mdr@science.uva.nl

Abstract. In this paper we address the task of finding topically relevant email messages in public discussion lists. We make two important observations. First, email messages are not isolated, but are part of a larger online environment. This context, existing on different levels, can be incorporated into the retrieval model. We explore the use of thread, mailing list, and community content levels, by expanding our original query with term from these sources. We find that query models based on contextual information improve retrieval effectiveness. Second, email is a relatively informal genre, and therefore offers scope for incorporating techniques previously shown useful in searching user-generated content. Indeed, our experiments show that using query-independent features (email length, thread size, and text quality), implemented as priors, results in further improvements.

1 Introduction

An archived discussion list records the conversations of a virtual community drawn together by a shared task or by a common interest [22]. Once subscribed, people are able to receive and send emails to this list. Most mailing lists focus on a fairly narrow domain to allow for more in-depth discussion among the participants, and as such, often serve as a general reference about the subject matter. To make this information accessible, effective tools are needed for searching in mailing list archives.

In this paper, we focus on one task: finding topically relevant messages in an email archive. From a retrieval point of view, this task presents some unique challenges. We limit ourselves to the following: (1) Email messages are not isolated. Being either an initial message or a response, they are part of a conversation (thread). Similarly, the mailing list itself is not an island, but part of a larger online environment. Can we make use of this contextual information and incorporate it into the retrieval model? (2) Email is a relatively informal genre, and therefore offers scope for incorporating techniques previously shown useful in user-generated content. Do counterparts of these methods exist in the domain of email search? If so, does their usage affect retrieval performance?

We explore these questions using the archived World Wide Web Consortium (W3C) mailing lists that were the focus of the email search task in 2005 and 2006 at the Enterprise track of the Text Retrieval Conference (TREC).

Specifically, to address (1), we first identify five context levels, and then explore the use of thread, mailing list, and community content levels in detail. We make use of these sources by expanding the original query with terms from these sources. To address (2), we take collection characteristics previously shown useful in user-generated content (in particular: blogs) and introduce their counterparts for email search. This results in three query-independent features: email length, thread size, and text quality.

M. Boughanem et al. (Eds.): ECIR 2009, LNCS 5478, pp. 400–411, 2009.

We employ a language modeling approach, for two reasons. First, our baseline model delivers very competitive performance, compared to participants of the TREC Enterprise track. Second, language models provide a theoretically sound framework for incorporating contextual factors in the form of query models and query-independent features in forms of document priors.

Our analysis reveals that query models based on contextual information can improve email archive search effectiveness over a strong baseline, but that the gains differ across topic sets and sources of contextual information. As to priors, we find that, on top of the improvements delivered by our query models, they improve even further, and they do so across the board.

The remainder of the paper is organized as follows. We discuss related work in the next section. Then, we detail our experimental setup and baseline retrieval approach. We continue with a set of experiments around query models and a set of experiments around prior information before concluding with a brief discussion.

2 Related Work

Research on email has traditionally focused on tools for managing personal collections, in part because large and diverse collections were not available for research use [9]. Triggered by the introduction of the Enron [12] and W3C [30] collections, opportunities opened up to study new challenges. A large body of these efforts focused on people-related tasks, including name recognition and reference resolution [7, 19, 20], contact information extraction [1, 5], identity modeling and resolution [9], discovery of peoples' roles [16], and finding experts [1, 25, 33]. Another line of work centers around efficient access to email-based discussion lists. Tuulos et al. [29] introduce a system that provides access to large-scale email archives from multiple viewpoints, using faceted search. Newman [22] explores visualization techniques to aid the coherent reading of email threads. Following this line of work, a number of research groups explored email search as part of the TREC 2005 [4] and 2006 [26] Enterprise tracks. Common approaches include the use of thread information to do document expansion, the use of filters to eliminate non-emails from the collection, assigning different weights to fields in emails (ads, greetings, quotes, etc), and smoothing the document model with a thread model.

One can view email as user-generated content: after subscribing to a mailing list, users are free to send whatever they want to the list, without an editor stopping them. In a way communicating through a mailing list is comparable to blogging: it is one-to-many communication, readers have the possibility to respond (email or comments), there are no rules on what to write, and both have a similar structure (blog-posts-comments vs. thread-mails-quotes). Within blog (post) search, the TREC Blog track [23, 17] plays an important role; having started in 2006, many approaches to blog post finding have been deployed. Among these approaches are the use of credibility indicators [31], recency and link structure [21], and query expansion on external corpora [32, 8].

An important part of trying to find most relevant documents is taking into account the various *aspects* of a given query [3]. To improve "aspect recall" we can use query modeling, i.e., transformations of keyword queries into more detailed representations of an information need (e.g., by assigning (different) weights to terms, expanding the query, or using phrases). Most approaches use the top retrieved documents as examples

from which to select terms to improve the retrieval performance [24]. In the setting of language modeling approaches to query expansion, we can estimate additional query language models [14, 28] or relevance models [15] from a set of feedback documents. Various ways of improving aspect recall have been introduced: Kurland et al. [13] provide an iterative "pseudo-query" generation technique to uncover multiple aspects of a query, using cluster-based language models. Weerkamp and de Rijke [32] explore the use of external corpora to uncover multiple viewpoints on a topic, an approach similar to [8]. Recently, the issue of aspect recall has been addressed using example documents, provided by the user, from which new query terms are sampled [2].

3 Experimental Setup

To answer the research questions identified in the introduction, we run a number of experiments, under the conditions listed below.

Dataset. The test collection we use is the *lists* part of the W3C collection [30]. This comprises 198,394 documents, however, not all of these are actual email messages, some of them are navigational pages. We use a cleaned version of the corpus by Gianluca Demartini (with navigational pages removed) and we use thread structure contributed by W3C[1]. After processing the thread structure we end up with 30,299 threads.

As an external corpus for query modeling purposes, we use the *www* part of the W3C corpus, consisting of 45,975 documents.

We use the topic sets developed for the Discussion Search (DS) task: 59 topics from 2005 and 50 topics from 2006. For all our runs we use only the title field of the topics and ignore all other information available (e.g., narrative or description). Relevance assessments for the DS task come on multiple levels. For this paper we focus on the *topical relevance* of documents (emails); experiments in other domains (e.g. blogs [17]) show that a strong baseline is most important in finding documents that fulfill additional constraints (i.e. opinionated in blog, with discussion in this case).

Evaluation Metrics. The measures we report at are Mean Average Precision (MAP), precision after 5 and 10 documents retrieved (P@5 and P@10, respectively), and Mean Reciprocal Rank (MRR).

Significance testing. For determining significance of differences between runs, we use a two-tailed paired T-test and report on significant differences using $^{\triangle}$(and $^{\triangledown}$) for $\alpha = .05$ and $^{\blacktriangle}$(and $^{\blacktriangledown}$) for $\alpha = .01$.

4 Our Baseline Retrieval Approach

We use a standard language modeling approach for our baseline system. In this query likelihood approach, documents are ranked according to the likelihood of being relevant given the query: $P(D|Q)$. Instead of calculating this probability directly, we apply Bayes' rule, then drop $P(Q)$ as it does not affect the ranking of documents. This leaves us with $P(D|Q) \propto P(D) \cdot P(Q|D)$.

[1] http://ir.nist.gov/w3c/contrib/

Assuming that query terms are independent from each other, we estimate $P(Q|D)$ by taking the product across terms in the query. We obtain

$$P(D|Q) \propto P(D) \cdot \prod_{t \in Q} P(t|D)^{n(t,Q)}. \quad (1)$$

Here, $n(t, Q)$ is the number of times term t is present in the query Q. This is the multinomial view of the document model, i.e., the query Q is treated as a sequence of independent terms [18, 27, 10].

Next we rewrite this Eq. 1 in the log domain, and generalize $n(t, Q)$ so that it can take not only integer but real values. This will allow more flexible weighting of query terms. We replace $n(t, Q)$ with $P(t|\theta_Q)$, which can be interpreted as the weight of term t in query Q. We will refer to θ_Q as *query model*. We generalize $P(t|D)$ to a *document model*, $P(t|\theta_D)$, and arrive at our final formula for ranking documents:

$$\log P(D|Q) \propto \log P(D) + \sum_{t \in Q} P(t|\theta_Q) \cdot \log P(t|\theta_D). \quad (2)$$

Three components still need to be defined: *document prior*, *document model* and *query model*. In the baseline setting we set $P(D)$ to be uniform. In Section 6 below we detail alternative ways of setting $P(D)$ based on insights from search in user-generated content. The document model is defined as $P(t|\theta_D) = (1-\lambda) \cdot P(t|D) + \lambda \cdot P(t|C)$, where we smooth the term probability in the document by the probability of the term in the collection. We use Dirichlet smoothing and set $\lambda = \frac{\beta}{\beta + |D|}$, where $|D|$ is the length of document D and β is a parameter; we set β to be the average document length (i.e., 190 words in email search). Both $P(t|D)$ and $P(t|C)$ are calculated similar to the baseline query model $P(t|\theta_Q)$:

$$P(t|\theta_Q) = P(t|Q) = \frac{n(t, Q)}{\sum_{t'} n(t', Q)}, \quad (3)$$

where $n(t, Q)$ is the frequency of term t in Q. In the following section we explore other possibilities of estimating the query model θ_Q.

5 Query Models from Email Contexts

In this section we consider several ways of expanding the baseline query model introduced in the previous section. To motivate our models, we start from the following observation. Emails are not just isolated documents, but are part of a larger online environment. This becomes apparent at different levels:

Sub-email level: Many of the emails sent to a mailing list are a reply on a previous message. Nettiquette dictates that when replying to an email, one should include the relevant part of the original email (as quote) and write one's response directly below this quoted text. Emails are not simply flat documents, but contain quotes, that may go back several rounds of communication. In this section we do not explore the possibilities of using this sub-email (re)construction, but in Section 6 we will shortly touch on it.

Thread level: One level above the actual email, we find the thread level. In mailing lists, emails concerning the same topic (i.e., replies that go back to the same originating email) are gathered in a thread. This thread is the "full" conversation, as

recorded by the mailing list. The content of the thread is the direct context in which a specific email is produced and could therefore offer very topic and collection specific information on the individual email. We explore this level further in the remainder of this section.

Mailing list level: This is the collection of all email messages and threads, in other words, the whole discussion list. This level serves as a context to all conversations and represents the general language usage across the mailing list. We make use of this information later in this section.

Community content level: The mailing list itself is usually part of a larger online community: the mailing list is the way to communicate with community members, but additional information on the community might be available. For the data set we use in this paper, the mailing list is accompanied by a web site (referred to as "w3c-www"). Information on the pages of this site are most likely related to topics discussed on the mailing list and we are therefore interested in using this information in the process of retrieving emails.

Community member level: The final level we discuss here is the level of community members: a community would not have content if it was not for the members of a community. The emails in mailing lists offer direct insight in which members are active (i.e., contributing a lot to the list), which roles different members have (e.g., always asking, always the first to answer, etc.), and what other content they have produced. Connecting emails to people, people to other people, and people to additional content (e.g., web pages) we can potentially extract additional information regarding the emails. However, this level of the environment is not further discussed in this paper, because it is not likely to have any impact on "plain" (topical) email search.

In this paper we explore the use of thread, mailing list, and community content levels. We expect the language used in community content (i.e., on W3C web pages) to reflect the technical nature of the topics. Similarly, language associated with the actual communications of members are represented in the mailing list, and language associated with discussion on a certain topic is represented in the threads. An obvious way of using these three sources is by expanding our original query with terms from either of these sources; to this end we employ the models introduced by [6] and [15].

5.1 Modeling

One way of expanding the original query is by using blind relevance feedback: assume the top M documents to be relevant given a query. From these documents we sample terms that are used to form the expanded query model $\hat{Q}$. Lavrenko and Croft [15] suggest a reasonable way of obtaining $\hat{Q}$, by assuming that $P(t|\hat{Q})$ can be approximated by the probability of term t given the (original) query Q. We can then estimate $P(t|\hat{Q})$ using the joint probability of observing t together with the query terms $q_1, \ldots, q_k \in Q$, and dividing by the joint probability of the query terms:

$$P(t|\hat{Q}) \approx \frac{P(t, q_1, \ldots, q_k)}{P(q_1, \ldots, q_k)} = \frac{P(t, q_1, \ldots, q_k)}{\sum_{t'} P(t', q_1, \ldots, q_k)}.$$

In order to estimate the joint probability $P(t, q_1, \ldots, q_k)$, Lavrenko and Croft [15] propose two methods that differ in the independence assumptions that are being made;

here, we opt for their relevance model 2 (RM2) as empirical evaluations have found it to be more robust and to perform slightly better. We assume that query words $q_1, \ldots, q_k$ are independent of each other, but we keep their dependence on t:

$$P(t, q_1, \ldots, q_k) = P(t) \cdot \prod_{i=1}^{k} \sum_{D \in M} P(D|t) \cdot P(q_i|D). \quad (4)$$

That is, the value $P(t)$ is fixed according to some prior, then the following process is performed k times: a document $D \in M$ is selected with probability $P(D|t)$, then the query word q_i is sampled from D with probability $P(q_i|D)$.

We used RM2 in three ways. One is where the documents $D \in M$ are taken to be email messages. The second is where they are taken to be the email threads in the W3C corpus. The third is where they are taken to be the WWW part of the W3C corpus (as described in Section 3). These three methods correspond to query expansion on the mailing list, thread, and community content levels, respectively.

Parameter estimation. For the models just described we need to set a number of important parameters: M, the number of feedback documents, K, the number of selected terms from the top M documents, and λ, the weight of the original query. To estimate them, we train on one year of our data set and test on the other year. The best settings for query modeling using the mailing list are: $\lambda = 0.7$, $M = 5$, $K = 5$. The best settings for query modeling using threads are: $\lambda = 0.6$, $M = 15$, and $K = 5$. The best settings for query modeling using w3c-www are: $\lambda = 0.8$, $M = 5$, and $K = 5$.

5.2 Results

The results for our baseline (Eq. 3) and expanded runs are listed in Table 1; the expansions considered are against the mailing list itself ("mailing list"), the WWW part of the W3C corpus ("w3c-www") and a corpus consisting of email threads ("threads"). The baseline performance is competitive; at the 2005 edition of the TREC Enterprise track the baseline run would have ranked in the top 3; for 2006 its performance would have been above the median [4, 26]. We see that expansion against the mailing list, against WWW documents, and against email threads all improve retrieval performance in terms of MAP, but there is no clear winner. Gains in terms of MAP are modest for 2006 and significant for 2005. For early precision measures (P@5, P@10, MRR) a mixed story emerges, as is to be expected: in some cases expansion hurts early precision, in others it improves. However, apart one case (2005 topics, expansion against threads, MRR) the differences are not statistically significant.

5.3 Analysis

In the previous subsection we explored the use of the context of the emails to improve email retrieval performance. Results show that certain aspects of the context could be used to improve performance. More specifically, we use information available in the mailing list, in an email's thread, and in the W3C web pages to enrich the original query. Besides the raw numbers, we are interested in a more detailed analysis of what happens when using these contextual factors. Figure 1 shows the comparisons on AP per topic between the non-expanded baseline and each expanded run and gives an idea of how many topics benefit from using context in query expansion.

Table 1. Results for baseline approach, expansion on mailing list, w3c-www, and threads for 2005 and 2006 topics

	2005				2006			
Level	**MAP**	**P@5**	**P@10**	**MRR**	**MAP**	**P@5**	**P@10**	**MRR**
-	0.3522	0.6000	0.5492	0.7481	0.3541	0.5960	0.5720	**0.7438**
mailing list	0.3743$^{\Delta}$	0.5932	0.5627	0.7669	**0.3636**	**0.6200**	**0.5760**	0.7252
w3c-www	0.3535	0.5864	0.5220	0.7815	0.3627	0.5800	0.5700	0.7372
threads	**0.3818**$^{\Delta}$	**0.6237**	**0.5712**	**0.7945**$^{\Delta}$	0.3624	0.5760	0.5500	0.6972

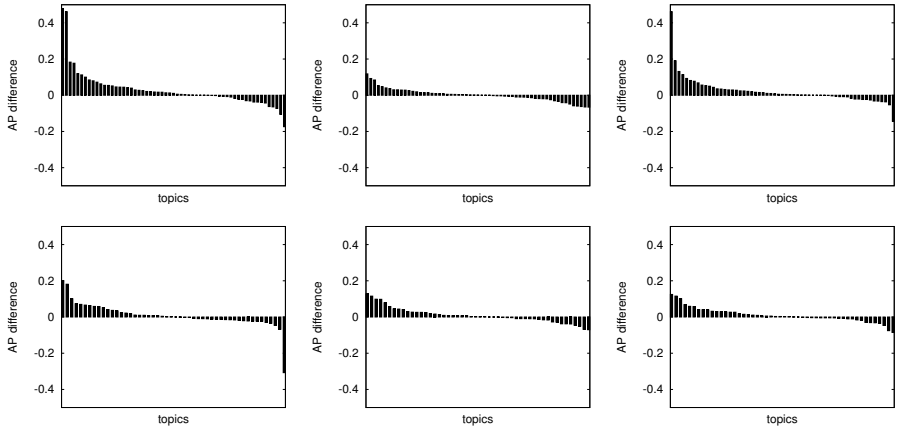

Fig. 1. Per-topic comparison between expanded runs and baselines for (Top) 2005 and (Bottom) 2006; (Left): threads. (Middle) w3c-www. (Right): mailing list.

We identify several interesting topics: topic 97 (*evaluation of color contrast*) for example shows a rather large drop when expanding on the web pages, but shows the largest improvement when expanded on the threads. The nature of this topics seems rather non-technical, or at least not so much related to W3C. Another topic that shows this behavior is topic 15 (*change chiper spec*): it gets a huge boost from expanding on threads, but drops when expanded on the web pages. One likely cause for this is the language usage in the query (e.g. "specs"), this is more similar to unedited language (as in emails) than to edited language. In general we see that queries that are rather specific have a better chance of getting a boost from expansion on the W3C web pages (e.g. "VoiceXML", "SOAP headers", "P3P"). Besides that, the main reason for topics failing on this expansion corpus is in both the broadness of topics (e.g. *devide independence*, *privacy cookies*) and in the less technical, W3C-related nature of the topics (e.g. *blocking pop-ups*).

A final part of our analysis is exploring the number of unique documents retrieved by one run compared to the others; we check how many relevant documents are present in a run A and not in runs B, C, and D. This is done for each run. The results of our comparisons are listed in Table 2.

From the results in the table we observe that each run introduces several new relevant emails that the other runs do not return. As we expected the different contextual levels capture different viewpoints on the topics and introduce each their own set of relevant results.

Table 2. Number of unique relevant results for each runs

year	baseline threads	w3c-www	mailing list
2005	20	42	18	7
2006	41	104	47	30

6 The Importance of Prior Information

Previous work on searching semistructured document collections [11] and on searching user-generated content [31] has revealed that using priors in the retrieval process can improve retrieval effectiveness. In this section we introduce three (groups of) priors that we believe can have a positive effect on retrieval performance: *email length*, *thread size*, and *text quality*.

6.1 Modeling

Email length. In Section 5 we already mentioned the sub-email level: emails do not only contain text written by the sender of the email, but also quoted text from previous emails. We hypothesize that using email length as a prior leads to improvements in retrieval effectiveness: people that have more valuable insight in a topic require more text to convey their message. Since we are interested in text generated by the actual sender of the email, we ignore quoted text. We touch on the sub-email level by removing content identified as quotes and estimate our email length prior on the non-quoted text: $P(D) = \log(|D|)$.

Thread size. Here, we build on the intuition that longer threads (on a given topic) are potentially more useful than shorter threads, and hence that email messages that are part of a more elaborate thread should be preferred over ones from shorter threads (on the same topic). We model this as follows: $P(D) = \log(|thread_D|)$ where $thread_D$ is the (unique) thread containing email message D and $|thread_D|$ is the length of the thread measured in terms of the number of email messages it contains.

Text quality. The third prior that we consider concerns the quality of the email messages, that is, of the language used in the body of the message (after removal of quotes). Building on [31], we looked at spelling errors (implemented as $P_{spelling} = \frac{n(errors,D)}{|D|}$, where $n(errors, D)$ is the number of misspelled words in document D), the relative amount of shouting (implemented as $P_{shout} = \frac{n(shouts,D)}{|D|}$, where $n(shout, D)$ is the number of fully capitalized words (with more than 5 characters) in document D), as well as the relative amount of emoticons (implemented as $P_{emoticons} = \frac{n(emo,D)}{|D|}$, where $n(emo, D)$ is the number of Western style emoticons in document D). Those three factors were multiplied to obtain a text quality prior, $P(D)$.

Combining priors. In some of our experiments we combined two or all three groups of priors. When combining email length and thread size, we take the average of the two values to be $P(D)$. Before adding the third prior, text quality, we normalize $P(D)$ by dividing each value by the maximum value for $P(D)$. After normalization, we take the average of the text quality prior and thread size-email length combination prior.

Table 3. Results for 2005 and 2006 topics: expanded baseline and (combinations of) priors

	2005				2006			
Prior	**MAP**	**P@5**	**P@10**	**MRR**	**MAP**	**P@5**	**P@10**	**MRR**
QMs from mailing threads								
-	0.3818	0.6237	0.5712	0.7945	0.3624	0.5760	0.5500	0.6972
(A) email length	0.3724	0.6034	0.5475	**0.8251**	0.3723	0.6080$^{\triangle}$	0.5820$^{\triangle}$	0.7276
(B) thread size	0.2990$^{\blacktriangledown}$	0.5593	0.4932$^{\blacktriangledown}$	0.7206	0.2729$^{\blacktriangledown}$	0.6280	0.5740	**0.8042**$^{\triangle}$
(C) text quality	0.3827	0.6305	**0.5729**	0.8057	0.3634	0.5960$^{\triangle}$	0.5560	0.6989
A + B	0.3789	**0.6407**	0.5559	0.8245	**0.3802**$^{\blacktriangle}$	**0.6320**$^{\blacktriangle}$	**0.5940**$^{\blacktriangle}$	0.7533$^{\triangle}$
A + B + C	**0.3903**$^{\blacktriangle}$	**0.6407**	0.5644	0.8176	0.3753$^{\blacktriangle}$	0.6120$^{\triangle}$	0.5780$^{\blacktriangle}$	0.7208
QMs from w3c-www								
-	0.3535	0.5864	0.5220	0.7815	0.3627	0.5800	0.5700	0.7372
(A) email length	0.3488	0.6102	0.5203	0.8038	0.3652	0.6080	0.5860	0.7577
(B) thread size	0.2772$^{\blacktriangledown}$	0.5424	0.4881	0.6721$^{\triangledown}$	0.2735$^{\blacktriangledown}$	0.6240	0.5780	**0.7861**
(C) text quality	0.3531	0.5932	0.5237	0.7784	0.3631	0.5840	0.5620$^{\triangledown}$	0.7310
A + B	0.3521	**0.6136**	**0.5390**	**0.8161**	**0.3745**$^{\triangle}$	**0.6400**$^{\blacktriangle}$	**0.5940**	0.7534
A + B + C	**0.3600**$^{\blacktriangle}$	0.5966	0.5322	0.7920	0.3723$^{\blacktriangle}$	0.6160$^{\triangle}$	0.5700	0.7394
QMs from mailing list								
-	0.3743	0.5932	**0.5627**	0.7669	0.3636	0.6200	0.5760	0.7252
(A) email length	0.3635	**0.6068**	0.5508	**0.7784**	0.3699	0.6560$^{\triangle}$	0.5900	0.7499
(B) thread size	0.2945$^{\blacktriangledown}$	0.5797	0.5000$^{\blacktriangledown}$	0.6989	0.2663$^{\blacktriangledown}$	**0.6800**$^{\triangle}$	0.5780	**0.7909**
(C) text quality	0.3748	0.5932	0.5610	0.7663	0.3638	0.6200	0.5740	0.7240
A + B	0.3697	**0.6068**	0.5508	**0.7784**	**0.3761**$^{\blacktriangle}$	0.6520$^{\triangle}$	**0.5960**	0.7555$^{\triangle}$
A + B + C	**0.3793**	0.5864	0.5525	0.7658	0.3738$^{\blacktriangle}$	0.6360	0.5840	0.7334

6.2 Results

Looking at the results listed in Table 3 we see slightly different stories for 2005 and 2006: on the 2005 topics the runs using all priors combined perform best in terms of MAP, and in case of mailing threads and w3c-www it performs significantly better than their counterparts without priors. For the other metrics the image is mixed, although in general the email length+thread size prior performs best in terms of early precision and MRR. For the 2006 topics the results are slightly different: highest scores on most metrics are obtained by using the email length+thread size prior, although differences with the combination of all priors are only marginal. For MRR the thread size prior performs best in all cases.

Looking at the three levels of contextual information, we see that query models constructed from mailing threads perform best. The difference between the runs using web pages (w3c-www) and emails are marginal in case of 2006 topics; for 2005 topics emails get a higher MAP score, but the improvement is not significant. The w3c-www query models on the other hand do improve significantly.

6.3 Analysis

We look in more detail at the results obtained in the previous section. From a first glance, the most interesting prior is the thread size prior: First, its performance on the 2006 topics is remarkable. Although MAP is significantly lower than the baseline, performance on early precision (P@5) and especially on MRR is very good. Using the thread size as

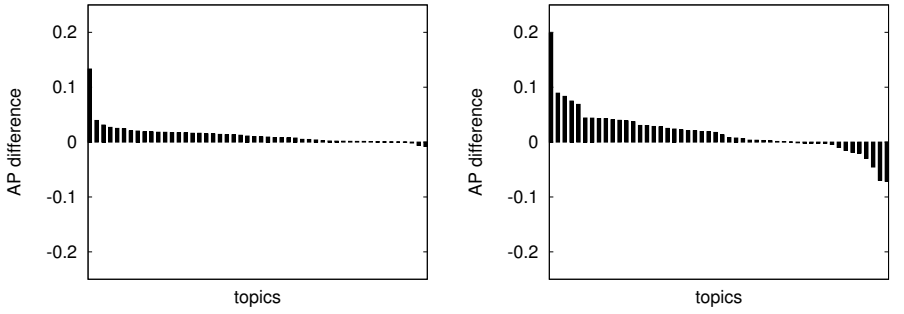

Fig. 2. Per-topic comparison between thread-expanded run without priors and . (Left): with all priors combined. (Right): thread size and email length combined.

a prior pushes relevant emails to the top of the ranking, but also causes recall to drop. Interesting to see is the combination between thread size and email length. Even though the MAP performance of the thread size prior is much lower than the performance of email length as prior, the combination of the two performs in all cases better than each of the priors individually. An email that contains a fair amount of newly generated text and that is part of a longer email discussion proves to have a higher chance of being relevant.

The strength of all our selected priors is shown when they are combined. The combination of thread size, email length, and text quality priors delivers a solid performance: In all cases the improvement in MAP over the expanded runs without priors is significant for $\alpha = .01$. When we zoom in on the thread-expanded runs on the 2006 topics, we see the highest MAP achieved by email length+thread size. Still, the improvement over the baseline by the combination of all priors has a higher confidence level, indicating the improvement is valid for more topics. Indeed, Figure 2 shows that almost all topics improve using the combination of priors (Left), whereas for the combination of thread size and email lenght (Right) more topics show a drop in AP.

7 Conclusions

In this paper we addressed the task of finding topically relevant messages in a public email archive. The main contribution of the paper is two-fold. First, we argue that email messages are not isolated but are part of a larger online environment. We identify a number of context levels and demonstrate that contextual information (in particular: thread, mailing list, and community content levels) can improve retrieval effectiveness. Second, since email is an informal genre, we investigate the effect of using collection characteristics previously shown useful in user generated content (in particular: blogs). We find that these query-independent features (namely: email length, thread size, and text quality) result in further improvements. Our approach for retrieval employs language models, which provide a theoretically sound framework to incorporate the above contextual factors in the form of query models and document priors, respectively. For experimental evaluation we use the W3C collection and email search topics from the 2005 and 2006 editions of the TREC Enterprise track.

Given this work, a natural follow-up is to enhance topical search with another criteria: opinions and arguments. This task, referred to as *discussion search* at the TREC Enterprise track, is defined as follows: identify emails that contribute at least one statement in favor of or against a specified topic (that is, identifying at least one pro or con argument about the topic). An obvious starting point would be to apply methods devised for finding opinions in user-generated content (again: blogs). Email archives, however, open up unique opportunities as well; participants are uniquely identified by their email address, and (in case of the W3C collection) some of them are also part of the organization to which the mailing list belongs. One could, therefore, look at individuals and their behavior, and try to leverage information from this additional context layer (referred to as *community member level* in the paper) into the retrieval model.

Acknowledgments

This research was supported by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments (http://www.stevin-tst.org) under project number STE-09-12, and by the Netherlands Organisation for Scientific Research (NWO) under project numbers 220-80-001, 017.001.190, 640.001.501, 640.002.501, 612.066.512, 612.061.814, 612.061.815, 640.004.802.

References

[1] Balog, K., de Rijke, M.: Finding experts and their details in e-mail corpora. In: WWW 2006 (2006)
[2] Balog, K., Weerkamp, W., de Rijke, M.: A few examples go a long way. In: SIGIR 2008, pp. 371–378 (2008)
[3] Buckley, C.: Why current IR engines fail. In: SIGIR 2004, pp. 584–585 (2004)
[4] Craswell, N., de Vries, A., Soboroff, I.: Overview of the TREC-2005 Enterprise Track. In: The Fourteenth Text REtrieval Conf. Proc. (TREC 2005) (2006)
[5] Culotta, A., Bekkerman, R., Mccallum, A.: Extracting social networks and contact information from email and the web. In: CEAS-1 (2004)
[6] Diaz, F., Metzler, D.: Improving the estimation of relevance models using large external corpora. In: SIGIR 2006, pp. 154–161 (2006)
[7] Diehl, C.P., Getoor, L., Namata, G.: Name reference resolution in organizational email archives. In: SIAM Int. Conf. Data Mining 2006, pp. 20–22 (2006)
[8] Elsas, J.L., Arguello, J., Callan, J., Carbonell, J.G.: Retrieval and feedback models for blog feed search. In: SIGIR 2008, pp. 347–354 (2008)
[9] Elsayed, T., Oard, D.W.: Modeling identity in archival collections of email: A preliminary study. In: CEAS 2006, pp. 95–103 (2006)
[10] Hiemstra, D.: Using Language Models for Information Retrieval. PhD thesis, University of Twente (2001)
[11] Kamps, J., de Rijke, M., Sigurbjörnsson, B.: The Importance of Length Normalization for XML Retrieval. Information Retrieval 8(4), 631–654 (2005)
[12] Klimt, B., Yang, Y.: Introducing the enron corpus. In: Conference on Email and Anti-Spam (2004)
[13] Kurland, O., Lee, L., Domshlak, C.: Better than the real thing?: Iterative pseudo-query processing using cluster-based language models. In: SIGIR 2005, pp. 19–26 (2005)
[14] Lafferty, J., Zhai, C.: Probabilistic relevance models based on document and query generation. In: Language Modeling for Information Retrieval. Springer, Heidelberg (2003)

[15] Lavrenko, V., Croft, W.B.: Relevance based language models. In: SIGIR 2001, pp. 120–127 (2001)
[16] Leuski, A.: Email is a stage: discovering people roles from email archives. In: SIGIR 2004, pp. 502–503. ACM, New York (2004)
[17] Macdonald, C., Ounis, I., Soboroff, I.: Overview of the trec 2007 blog track. In: TREC 2007 Working Notes, pp. 31–43 (2007)
[18] Miller, D., Leek, T., Schwartz, R.: A hidden Markov model information retrieval system. In: SIGIR 1999, pp. 214–221 (1999)
[19] Minkov, E., Wang, R.C., Cohen, W.W.: Extracting personal names from emails. In: HLT-EMNLP 2005 (2005)
[20] Minkov, E., Cohen, W.W., Ng, A.Y.: Contextual search and name disambiguation in email using graphs. In: SIGIR 2006, pp. 27–34 (2006)
[21] Mishne, G.: Applied Text Analytics for Blogs. PhD thesis, University of Amsterdam (2007)
[22] Newman, P.S.: Exploring discussion lists: steps and directions. In: JCDL 2002, pp. 126–134. ACM, New York (2002)
[23] Ounis, I., Macdonald, C., de Rijke, M., Mishne, G., Soboroff, I.: Overview of the TREC 2006 Blog Track. In: TREC 2006. NIST (2007)
[24] Rocchio, J.: Relevance feedback in information retrieval. In: The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice-Hall, Englewood Cliffs (1971)
[25] Schwartz, M.F., Wood, D.C.M.: Discovering shared interests using graph analysis. Commun. ACM 36(8), 78–89 (1993)
[26] Soboroff, I., de Vries, A.P., Craswell, N.: Overview of the trec 2006 enterprise track. In: The Fifteenth Text REtrieval Conference Proceedings (TREC 2006) (2007)
[27] Song, F., Croft, W.B.: A general language model for information retrieval. In: CIKM 1999, pp. 316–321 (1999)
[28] Tao, T., Zhai, C.: Regularized estimation of mixture models for robust pseudo-relevance feedback. In: SIGIR 2006, pp. 162–169 (2006)
[29] Tuulos, V.H., Perkiö, J., Tirri, H.: Multi-faceted information retrieval system for large scale email archives. In: SIGIR 2005, pp. 683–683 (2005)
[30] W3C. The W3C test collection (2005), `http://research.microsoft.com/users/nickcr/w3c-summary.html`
[31] Weerkamp, W., de Rijke, M.: Credibility improves topical blog post retrieval. In: ACL 2008: HLT, pp. 923–931 (June 2008)
[32] Weerkamp, W., de Rijke, M.: Looking at things differently: Exploring perspective recall for informal text retrieval. In: DIR 2008, pp. 93–100 (2008)
[33] Zhang, J., Ackerman, M.S.: Searching for expertise in social networks: a simulation of potential strategies. In: GROUP 2005, pp. 71–80 (2005)