Leveraging Collection Structure in Information Retrieval With Applications to Search in Conversational Social Media

Ph.D. Thesis Proposal

Jonathan Elsas
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

May 1, 2009

Thesis Committee:
Jaime G. Carbonell, chair
Jamie Callan
William Cohen
Susan Dumais, Microsoft Research
Abstract

In this thesis, we propose to study methods of leveraging collection structure in document ranking algorithms. Document collections often have strong relationships among items in the collection. Documents may belong to the same category, they may share the same author, or they may be messages in a common conversation thread. These relationships between documents, and the collection structure the relationships provide, can inform our understanding of the documents’ content. For example, it is often necessary to view messages in the context of a thread in order to understand how a message contributes to the overall dialog. It may be necessary to known the author of a document to establish that document’s credibility. Although this collection structure is critical for comprehension, it is rarely exploited in retrieval algorithms. We hypothesize that if a retrieval algorithm can leverage this collection structure, it may be possible to improve retrieval performance.

We will evaluate three aspects of collection structure through the course of this thesis work. First, we will investigate modeling the topical relationship among items of a collection when ranking those collections. For example, we may wish to favor elements of the collection that are more similar to the collection as a whole rather than those tangential to the collection’s main focus. We have already shown this to be an effective technique in blog feed search [26]. Second, we will investigate modeling author expertise with respect to the query when ranking documents from collections with many authors. In email ranking, for example, it may be beneficial to favor messages written by a subject-matter expert rather than someone with little prior experience. Finally, we will investigate the use of high-level classifications in document ranking algorithms. In the case of online forums, for example, we may wish to favor messages from a related subforum when responding to a user’s query.

We propose extensions to two retrieval frameworks in order to integrate this collection structure into ranking algorithms. First, we propose a new probabilistic retrieval model based on a mixture-of-trees Bayesian network. This model is capable of differentially weighting the influence of different aspects of the collection structure, allowing fine-tuning of the influence of author expertise, for example, in the final document ranking. Second, we also propose to explore these aspects of collection structure in the learning-to-rank setting, where machine learning algorithms are used to fine-tune feature weights in the ranking algorithm. In this setting, the focus of the proposed work is on feature engineering, developing query-specific features that reflect the influence of different aspects of collection structure.

The extensions we propose generalize across many types of collection structure and ranking tasks. We hypothesize that application of these approaches will significantly improve the quality of search results and lend insight into methods for modeling things like author expertise.
## Contents

1 **Introduction**  
   1.1 Structure in Social Media Collections .............................................. 1  
   1.2 Information Retrieval Challenges ..................................................... 2  
   1.3 Thesis Statement .......................................................... 3  

2 **Related Work**  
   2.1 Language Modeling for Information Retrieval ........................................ 4  
   2.2 Using Document Attributes in Ranking ............................................. 4  
   2.3 Collections of Collections ............................................................ 5  
      2.3.1 Cluster-Based Retrieval ......................................................... 5  
      2.3.2 Resource Ranking ................................................................. 6  
      2.3.3 Expert search ................................................................. 7  
      2.3.4 Blog Feed Search .............................................................. 8  
   2.4 Information Retrieval for Social Media ............................................... 9  
      2.4.1 Email and Newsgroups ......................................................... 9  
      2.4.2 Online Message Boards and Forums ......................................... 10  
      2.4.3 Community Question Answering ............................................ 10  

3 **Relevant Work Completed**  
   3.1 Blog Feed Search ........................................................................... 12  
      3.1.1 Probabilistic Retrieval Models for Feed Search ....................... 12  
      3.1.2 Large Document Model .......................................................... 13  
      3.1.3 Small Document Models ......................................................... 14  
      3.1.4 Retrieval Model Experiments ................................................. 16  
      3.1.5 Conclusion ............................................................................... 18  
   3.2 Online Forums Search ..................................................................... 18  
      3.2.1 The Structure of Online Forums ............................................ 19  
      3.2.2 Dataset Description ............................................................... 20  
      3.2.3 Retrieval Models for Forum Search ....................................... 25  
      3.2.4 Evaluation, Results & Analysis ........................................... 26  
      3.2.5 Conclusion ............................................................................... 28  

4 **Planned Thesis Work**  
   4.1 Retrieval Algorithm Development .................................................... 29  
      4.1.1 Collection Structure in the Probabilistic Retrieval Framework .... 30  
      4.1.2 Feature-Based Ranking Models and Collection Structure .......... 34  
   4.2 Collection Development .................................................................. 36  
      4.2.1 Limitations of Existing Test Collections .................................. 36  
      4.2.2 Document Collection .............................................................. 36  
      4.2.3 Queries ................................................................................... 37  
      4.2.4 Document Pooling ................................................................. 38  
      4.2.5 Relevance Assessment ............................................................ 39  
   4.3 Tasks and Test Collections ............................................................... 39  
   4.4 Prediction Tasks ............................................................................ 40  

5 **Tentative Schedule**  


1 Introduction

Text document collections are often referred to as “unstructured data”, but this characterization is frequently inaccurate. Many document collections, in fact, contain rich organizational structure. This structure is frequently present as document attributes or metadata, and often used to organize the documents for presentation. In the context of this work, organizational collection structure, or simply collection structure, refers to an explicit, typically manually generated, organization of the items in the collection\(^1\). Collection structure may be artifacts of the document creation (such as timestamps), or generated by the authors, editors, or even users of the collection (such as categories or tags). Structure is common in many types of collections, some of which have been used as information retrieval test collections for years:

- **Newswire** collections contain bylines, datelines and publication timestamps.
- **Email** collections contain sender and recipient addresses, timestamps, and frequently “in-reply-to” headers which identify other messages in the collection to which a messages responds.
- **Blog** collections contain post timestamps, author and comment information, and often tags or categories.

Organizational collection structure has proven to be useful in all forms of information access. Library catalogs have been organized by various classification schemes for centuries. In electronic environments, faceted interfaces provide access to several structural dimensions for navigation and filtering. This type of structural access has been shown to be useful in personal information collections [22], image search [79], and has been proposed for access to blog collections [31] and general web search [36].

Organizational structure is often critical for understanding various aspects of a document collection. For example, the organization of messages into threads is needed to understand the conversational context of each message. Document timestamps on newswire articles are necessary to disambiguate cyclical and recurring events. Authorship information may be required to establish an article’s credibility.

Although collection structure is useful and often critical for information access and comprehension, little research is aimed at leveraging collection structure in ranking algorithms. In the existing literature, when collection structure is used in ranking, typically only a single attribute is explored at a time and often that is done without regard to the information need presented in the query. In this work, we propose a unified and flexible approach to leveraging collection structure in several existing document ranking frameworks.

1.1 Structure in Social Media Collections

“Social media” collections provide especially rich structure that may be leveraged in ranking algorithms. Social media websites have recently become popular destinations for web users. Current figures from Alexa, a service that monitors internet traffic through toolbar usage statistics, suggest that sites geared towards social networking and content sharing such as

\(^1\)In contrast to these explicit collection structures, implicit collection structure, such as document clustering, is beyond the scope of this thesis.
Facebook\textsuperscript{2}, YouTube\textsuperscript{3} and MySpace\textsuperscript{4} are make up three of the top five most visited sites in the United States [3]. Other social media sites are ranked in the top twenty most popular web destinations include Wikipedia\textsuperscript{5}, a communally-edited encyclopedia, and Blogger\textsuperscript{6}, a free blogging service. In addition, a recent Pew research study found that 35% of all adults online use some form of online social networking site, 32% read blogs and 11% have created a blog [34].

Although many of these sites function as primarily a means to socialize online, some do contain large text archives that are of ongoing interest to information seekers. These collections, referred to here as conversational social media, often foster dialog among a large number of participants, usually organized into message threads, and often topically categorized or tagged. Online message boards, newsgroups and public email lists are all example of conversational social media that generate potentially large textual archives with contributions from many participants.

1.2 Information Retrieval Challenges

Through their rich organizational structure, conversational social media collections offer new an interesting challenges for information retrieval. First, in these collections the unit of retrieval is not fixed, but dependent on the task. Consider online forums, often organized into topical sub-forums, which in turn are organized into conversation threads of individual posts. Some information needs many only require a single post as a result, some require the context of the full conversation thread, and others may need to retrieve a pertinent sub-forum.

These collections often offer another orthogonal axis of organization — the author. In highly trafficked message boards and mailing lists, tens or hundreds of thousands of users with varying levels of expertise contribute to the conversation. One may wish to find subject matter experts to address a question to, or favor message threads with contributions from those more likely to know the answer.

The conversational nature of these collections also provides strong relations between documents in the collection. Messages are responses to previous messages in the thread, often answering questions, quoting previous messages, and in other ways continuing the conversation. When searching, we may want to integrate information from the conversational context of a message, rather than solely focusing on the message contents.

In this thesis we aim to tackle some of these challenges. We propose to study the use of collection structure in document ranking algorithms, specifically with applications to ranking in conversational social media. Through a unified approach to model collection structure in ranking algorithms, we will explore the role of author expertise, relationships among related elements of a collection, and the role of topical collection organization in document ranking.

\textsuperscript{2}http://facebook.com  
\textsuperscript{3}http://youtube.com  
\textsuperscript{4}http://myspace.com  
\textsuperscript{5}http://wikipedia.org  
\textsuperscript{6}http://blogger.com
1.3 Thesis Statement

In this thesis we are interested in leveraging the structure in document collections when ranking. Conversational social media archives, including blogs, public email archives and online forums, present a rich venue within which to study collection structure.

We hypothesize that by designing document ranking algorithms to take advantage of this collection structure, we can significantly improve retrieval performance in a variety of tasks. We propose retrieval models that take a unified view of collection structure, and through the application of these models we hypothesize that:

1. When ranking topically-related sub-collections of documents, modeling the relatedness of individual items to their sub-collections can lead to significant performance improvements. We have already shown this to be the case in blog feed search (see Section 3.1).

2. In collections with contributions from a variety of authors, modeling the author’s expertise in ranking algorithms will lead to performance improvements.

3. In collections with a high-level topical organization, such as a classification scheme, modeling the affinity of a topic to the query and to elements of the collection will also lead to performance improvements.
\section{Related Work}

Several branches of information retrieval research are directly applicable to this work. In this section we discuss previous work in these related areas, and present relevant models when informative with regard to the proposed work.

This section is organized as follows: First, we will give a brief overview of language modeling for information retrieval. Next, we will discuss previous work dealing generally with collection structure. That work which is most relevant to the proposed work here is previous work on retrieval of \textit{sub-collections} of documents, such as resource ranking in federated search, cluster-based retrieval, expert finding and blog feed search. Finally, we will present work relating to search in social media, including general approach to search in blogs and other social media.

\subsection{Language Modeling for Information Retrieval}

Statistical language models were proposed as a more principled alternative to TF-IDF vector space scoring models \cite{63}, and subsequent work in language modeling is the foundation for many of the ranking algorithms proposed here. In the language modeling framework, statistical term distributions are computed for both the query ($\theta_Q$) and the documents ($\theta_{D_i}$), such that $P(w|\theta)$ represents the likelihood of observing the word \textit{w} in the model $\theta$. Several methods have been proposed for the ranking of documents in the language modeling framework, and in this work we will build upon the \textit{query likelihood} model. In this model, documents are ranked by the likelihood of generating the query language model from the document’s, i.e. the query likelihood, $P(\theta_Q|\theta_{D_i})$. We will often refer to the document and query language models as simply $D$ and and $Q$ respectively.

The effective and efficient estimation of these models has been a central component to research in language modeling for information retrieval. Zhai and Lafferty investigated modeling the document as a multinomial, applying several smoothing methods to this estimation \cite{80}. Metzler et al. model the document as a multiple-Bernoulli distribution, allowing easier incorporation of non-term features into ranking, such as term windows or fields \cite{55}. Further work by Metzler and Croft relates the query likelihood estimation to estimating parameters in a Markov Random Field (MRF) model, combining term unigram, n-gram and term window features in the ranking function \cite{53}.

Throughout the proposed thesis work, we will make use of the multiple-Bernoulli and MRF retrieval models presented by Metzler and Croft \cite{53, 54, 55} and implemented in the Indri search engine\footnote{\url{http://www.lemurproject.org/indri}}.

\subsection{Using Document Attributes in Ranking}

Document attributes are often used in ranking algorithms as query-independent features that may indicate the overall quality of a document. In the language modeling framework, these are typically integrated into the document scoring via a \textit{document prior}, representing the prior probability of a document being relevant before observing the query. Features of hyperlinked documents, such as in-link count and PageRank are commonly used in web search to model the popularity or authority of a page \cite{57}. URL depth has been used as an indicator for whether a page is likely to be a homepage \cite{62}. Spam classification with
Several research studies have evaluated the use of document timestamps in information retrieval. Work by Li and Croft has shown that some query types are best served by recent documents [39]. Jones and Díaz have shown that the distribution of publication timestamps in the result set can be useful for identifying ambiguous and poorly performing queries [33]. This last study is particularly interesting due to its use of query-specific temporal features, although these features were not used in ranking algorithms, but rather for the purposes of precision prediction.

2.3 Collections of Collections

Collection structure is often present through organizations of documents along various dimensions. This, in turn, defines sub-collections of related documents, and a common IR task is to rank those sub-collections with respect to a user’s query.

The dimensions that define sub-collections often include:

- **Topicality**: Documents that share a topical focus, either explicitly defined through tagging or categorization, or implicitly through document clustering.

- **Reference**: Documents referring to the same entity, such as an employee in an enterprise setting.

- **Authorship**: Documents that share the same author.

- **Origin**: Documents residing in the same database, such as in a federated search setting.

This list is not meant to be an exhaustive enumeration of the possible ways to organize a collection, but rather highlight some examples of organizational structure that have been used in the past to define sub-collections for IR. Ranking sub-collections poses several challenges beyond just document ranking. In particular, when we rank sub-collections of documents we must consider how relevance at the document-level contributes to relevance at the collection-level.

Below we will discuss several specific areas of study that, at least in part, focus on retrieval of sub-collections. We draw inspiration from many of these established research areas in our proposed work.

2.3.1 Cluster-Based Retrieval

Cluster-based retrieval has been proposed to improve IR system performance and efficiency, and is typically viewed as a multi-stage process. This work is motivated by the Cluster Hypothesis, stating that documents relevant to the same query are likely to be similar to each other [70]. Thus, if a system can identify a cluster containing some documents relevant to the query, that cluster is likely to contain other documents also relevant to the query.

Cluster-based retrieval typically proceeds as follows: First, as a preprocessing stage, the document collection is divided into topically related, possibly overlapping clusters of documents. At query time, those clusters of documents most likely to contain relevant
documents are retrieved. Finally, documents are selected from within those clusters to form a final document ranking.

Work on clustering document collections and merging results from multiple clusters is beyond the scope of this work and will not be discussed here. Selecting or ranking clusters, however, is some of the first work dealing with retrieval of sub-collections of documents.

Early work on cluster-based retrieval by Salton [64] and van Rijsbergen [70] viewed the process primarily as a means to improve retrieval system efficiency. By selecting only a small number of document clusters to search, one need not evaluate the query against the entire collection. In their work, clusters are ranked by the degree of match between the query and a “cluster representative”, which may be a single document within the cluster or some summarized representation of the cluster contents such as the cluster centroid.

More recent work on cluster-based retrieval in the language modeling framework utilizes document clusters both for document selection, and as an additional source of evidence in the scoring of documents. This evidence has been utilized in several ways, for example as a means to smooth the document language model along with the collection [41], or to establish a notion of “centrality” with respect to the query [37] (or more precisely, with respect to the graph induced by a query-specific clustering). The latter method of using the contents of a cluster in ranking has a similar effect to pseudo-relevance feedback, leveraging the content highly ranked documents to re-score documents.

Liu and Croft explore a variety of cluster representations also in the language modeling framework [41, 42]. They find that two techniques improve cluster selection performance. The first is that language models derived form the individual elements of the cluster and then aggregated into an overall cluster score generally outperform language models based on treating the cluster as a single document. They also find that re-weighting or selectively choosing elements of the cluster, in particular to favor those likely to be relevant to the query, generally outperforms methods that treat each element of the cluster identically.

One of the Liu and Croft models, the Geometric mean, performs particularly well in their experiments and we use it as a baseline in several of our ranking experiments described in subsequent sections. This model ranks clusters by the query likelihood, estimated as follows:

\[
P(Q|\text{Cluster}) = \prod_{q \in Q} P(q|\text{Cluster})
\]

\[
= \prod_{q \in Q} \left( \prod_{1 \leq i \leq k} P(q|D_i) \right)^{\frac{1}{k}}
\]

where the \( i \) indexes the top-scoring \( D_i \) documents in the cluster with respect to the query \( Q \) and \( P(q|D) \) is a standard language modeling term score. This model effectively scores all the documents in a cluster, selects the top-\( k \) of those documents, and generates a final cluster score by taking the geometric mean of those top documents’ scores.

### 2.3.2 Resource Ranking

Distributed information retrieval, or federated search, is a set of tasks relating to document retrieval across many document collections rather than a single centralized index. One of those tasks, resource selection, is the ranking of available document collections in order to select those most likely to contain many documents relevant to a query.
Early work on resource ranking using belief networks by Callan et al. [16, 18] introduced the CORI ranking algorithm, which scores each collection based on summary statistics such as document frequencies and collection lengths. This algorithm assumes some knowledge of the collection statistics for each of the collections to be ranked. CORI ranks collections by their estimated query likelihood \( P(Q|Coll_i) \), where the likelihood of observing each query term \( q \) in a collection \( Coll_i \) is:

\[
P(q|Coll_i) = b + (1 - b) \times \frac{df}{df + 50 + 150 \times \frac{cw_i}{\text{avg}cw} \times \frac{\log(|DB|+0.5)}{\log(|DB|+1.0)}}
\]  

(3)

In the above formula, \( df_i \) are the number of documents in collection \( i \) that contain the query term, \( cf \) is the number of collections that contain the query term \( q \), \(|DB|\) is the number of collections, \( cw_i \) is the number of words in collection \( i \), \( \text{avg}cw \) is the average number of words per collection and \( b \) is the default belief. This ranking formula is based on TF-IDF weighting, assuming each collection can be represented by aggregate statistics such as the number of documents containing a query term.

Subsequent work by Si and Callan [67] introduced the ReDDE algorithm which models resources via the individual document they contain. This algorithm is based on the assumption that the collection statistics may not be readily available, and a sampling process must be performed to estimate the those statistics for each of the databases. ReDDE ranks collections based on their estimated number of relevant documents, which is calculated as follows:

\[
\text{Rel}_Q(i) = \sum_{d \in Coll_i} P(\text{rel}|d)P(d|Coll_i)N_i
\]

(4)

where \( P(\text{rel}|d) \) is a thresholded approximation of the probability of relevance of document \( d \), \( N_i \) is the number of documents in the \( i \)th collection, and \( P(d|Coll_i) \) is the likelihood of sampling document \( d \) from collection \( i \). Thus, this scoring formula models the relevance of individual elements within the sub-collection in order to estimate the relevance of the sub-collection. The ReDDE model, and others based on aggregating document scores have been shown to be superior to the “large document” approach used by CORI for resource selection in most situations.

These two models highlight a recurring theme in collection ranking: whether to represent collections as monolithic “large” documents, or to score the elements of the collection individually and translate those scores into a collection-wide score.

2.3.3 Expert search

Expert search has been studied in recent years in the Enterprise Track at the annual Text REtrieval Conference (TREC) [7, 21, 68]. The task of expert search is that of finding people within an organization with expertise pertinent to a user’s query. At TREC, this task has been performed with the use of an enterprise collection of documents, containing references to employee expert names or email addresses within the organization.

Two primary approaches to expert finding have emerged, and have been extensively studied and compared by Balog et al. [8]. These two approaches can be thought of as similar to the models of resource ranking and cluster selection described above. In the first of these models, referred to as Model 1 (also known as the “candidate model” or
“query independent approach”), textual representations of each candidate expert are built prior to query-time. These candidate profiles are typically constructed from the set of documents with references to the candidates names or email addresses. To construct the profile, a language model is created by “averaging” or concatenating of the documents in those sets. As in cluster-based retrieval, this forms a single representative pseudo-document for each candidate expert. When queries are issued into the system, candidates are ranked by the degree of match between the query and these pseudo-documents. The second model, referred to as Model 2 (also known as the “document model” or “query dependent approach”), documents are first ranked in a standard way with respect to the query. Then, this document ranking is translated into a candidate ranking by extracting the expert-document associations from those retrieved documents and applying some method of aggregating document scores into candidate scores.

More formally, these candidate ranking models have been presented by Balog et al. as follows [8]:

$$P(Q|ca) = \prod_{t \in Q} (\lambda P(t|ca) + (1 - \lambda)P(t|C))$$ Model 1, Candidate Model (5)

$$P(Q|ca) = \sum_d P(d|ca)P(Q|d)$$
$$= \sum_d P(d|ca) \prod_{t \in Q} (\lambda P(t|d) + (1 - \lambda)P(t|C))$$ Model 2, Document Model (6)

where $d$ are the documents in the collection, $ca$ are the candidate experts and $\lambda$ is a smoothing parameter and $C$ is a collection language model. In Model 1, the candidate language model $P(t|ca)$ is typically estimated by concatenating (or averaging) the document language models containing references to the candidate expert. In Model 2, a candidate language model is not formally defined, but rather a candidate ranking is derived from the document scores ($P(Q|d)$) and candidate associations ($P(d|ca)$). Model 2 has been shown to be superior in the expert finding task.

### 2.3.4 Blog Feed Search

Blog feed search is another task that has gained recent popularity based on ongoing research efforts at TREC. This is the task of ranking blogs with respect to a user query, where the returned blogs have a “principle, recurring interest” in the topic [47, 59, 60]. This is also a task of ranking sub-collections of documents, as the blogs are composed of a series of blog posts, typically written by the same author. This task has been viewed as similar to both expert finding [9, 46] and the resource selection [5, 26], and the existing approaches to blog feed search to some degree mirror approaches to those tasks.

Similarly to research in expert search and resource ranking, retrieval models that view blogs as a single “large document” have been compared to models that view blogs as a collection of individual “small documents”. Balog et al. directly applied their expert search models to the task of blog feed search [9]. They found that, in contrast to expert finding, Model 1 (the “large document” model) outperformed Model 2 (the “small document” model) for this task. Similar findings were shown by Seo and Croft [65, 66], Arguello et al. [5] and Elsas et al. [25], who all apply resource selection techniques to this task.

Subsequent studies by Elsas et al. [26], also described in Section 3.1, detail our approach to blog feed search, and its relation to other collection-ranking tasks. In these studies, we
show that when appropriately modeling the centrality of a blog post within the blog, significant performance improvements can be achieved.

2.4 Information Retrieval for Social Media

Social media collections pose new an interesting challenges for information retrieval research. As a form of social interaction, these collections are archives of dialog among the users. These collections can typically be organized along several dimensions, including general topic, discussion thread, and author. In this section, we will review previous research that looks at several social media collections that exhibit this type of structure. We focus here on email collections, online forums or message boards, and community question answer (CQA) archives.

Blog collections, described in the previous section, are often categorized as a social media. In those collections, the social interaction occurs in comments readers attach to a blog post, and via links across different blogs. Although we can consider blogs as a social media, approaches to blog search have largely ignored the social structure in the collection. This may be due to the difficulty in reliable distinguishing blog posts from comments at a large scale.

2.4.1 Email and Newsgroups

Retrieval with collections of email has also been studied at the TREC enterprise track in 2005 and 2006 [21, 68]. In these years, the enterprise track used a collection of archived email messages from the World Wide Web Consortium (W3C) [20] and several tasks were run, including known-item finding and discussion search. The W3C collection consists of several public email list archives, and messages within each of those lists are organized into message threads. Many of the approaches to the TREC tasks leverage this thread structure.

Several TREC participants took a fielded-retrieval approach, where email messages are modeled as a combination of text from the subject, body and quoted portions of the message. Ogilvie and Callan [58], working within the language modeling framework, modeled the email message as a mixture model, combining evidence from these sources as well as from response messages. Craswell et al. [20] took a similar approach using the BM25F ranking function. More recent work using the same collection by Weerkamp, et al. [75] leverages the language used at different levels of the thread context to influence which terms are use for query expansion. These different contexts are defined as the message itself, the message thread and the entire mailing list. All of these approaches found additional utility in leveraging email message priors, such as the depth of the message in the thread.

Other work on “search” in email collections by Minkov and Cohen [56] focuses on defining similarity measures between objects in the collection such as authors or messages. These similarity measures are then used to disambiguate person names or identify related messages, likely to belong to the same message thread. This work takes an alternate view of collection structure, rather than using the structure to identify sub-collections, it is used to define explicit relations between the objects, resulting in a graph-based representation of the collection.

Similar to email collections, newsgroups provide message, author and thread structure.
Xi et al. [76] looked at message retrieval in this setting, employing a learning-to-rank approach to combining various content and structural features in a ranking function. They found that textual features of the message, its thread, and other responses in the thread are the most useful in ranking.

The majority of the work in search over email or newsgroups is concerned with combining evidence from different parts of the message, such as the title, body or replies, as well as using non-textual features such as the number of replies. None of the previous work, however, is concerned with leveraging more complex collection structure, for example through modeling the expertise of the author when ranking messages. Additionally, all of these studies have looked at retrieving single messages, rather than a message thread.

2.4.2 Online Message Boards and Forums

Online messages boards support user discussion in a structured environment, with messages, threads, and frequently higher-level classifications of those threads. There are many thousands of public online forum websites and they typically cater to a specific topic of discussion such as health issues, movie reviews, programming languages or politics. Typically message boards contain several topical sub-forums, which in turn contain the message posts and post replies. The message post/response structure are displayed either flat or hierarchically and most frequently ordered according the time of posting. Message boards also frequently provide users with the ability to cultivate somewhat of a online personality with pictures and personal biographical profiles. Some message boards provide additional features such as galleries, recipe databases, explicit “social networking” features such as marking connections between other message board users.

There has been very little IR-related research dealing specifically with online forums. Several studies have looked at knowledge extraction in online forums, for example identifying question-answer pairs [19] or identifying responses that provide context and an answer to a previous question in the thread [23]. Somewhat similar to a retrieval task, Feng et al. [28] developed a “discussion-bot”, which responds to new forum posts with automatically identified related questions and answers. The question-matching component of this system retrieves likely answers with a vector-space TF-IDF ranking formula.

2.4.3 Community Question Answering

Community Question Answer (CQA) services such as Yahoo! Answers\(^8\) resemble online message boards in structural organization. On CQA sites, initial posts take the form of a question, and participants can post an answer in response. They are similar to online message boards in their organization into topical “threads” with contributions from different users. However, in CQA sites, users typically only provide a single answer in response to a question, whereas online forums provide more of a conversational structure, with users potentially contributing a series of posts responding to different messages in the thread at different times. CQA sites also have several features not present in online forums. On these sites, users accumulate “points” when their answers are judged best by the original asker, or voted on by others in the community.

Yahoo! Answers has been extensively studied by Agichtein et al. [1, 11, 12, 35, 38, 43, 44]. Their work explores various prediction tasks with this dataset, primarily focusing

\(^8\)http://answers.yahoo.com
on feature design and engineering. Several of the tasks studied include: automatically identifying “high quality” questions and answers [1], predicting if askers are satisfied by the answers provided [44] and identifying malicious voting in community judgement system [11].

Several studies have looked at CQA archives as data sources for factoid question answering. Bian et al. [12] used a Yahoo! Answers dataset and Xue et al. [77] use a similar dataset from Wondir\textsuperscript{9} for QA evaluations. In these studies, questions from several TREC QA datasets were used as queries and answers were identified from the TREC answer patterns [72, 73]. In the first study, answers were ranked via a learning-to-rank algorithm, using several content-based, community-based and statistical features from the dataset. In the second, a statistical translation model was adapted to retrieving questions and answers.

Some of these studies model answerer expertise in prediction or ranking. In the Agichtein studies, this is done through gross measures of expertise, rather than query- or question-specific measures. Some of the expertise measures used in those studies includes the number of “points” an answerer has received or several graph-based authority measures calculated over the user-question-answer graph [1]. This study reports that some of the most effective features in predicting the quality of an answer are community judgements on the previous answers provide by the same user.

\footnote{http://www.wondir.com}
3 Relevant Work Completed

To date, several aspects of collection structure have been investigated in the context of blog feed search [5, 26] and online forum search [24]. This section describes that completed work.

3.1 Blog Feed Search

Blog feed search is an information seeking task in which someone has an ongoing interest in a topic and plans to follow blogs discussing that topic on a regular basis, possibly through their feed reader. Several commercial blog search engines exist (blogsearch.google.com, search.live.com/feeds, bloglines.com/search, technorati.com/search). Most of these present a feed search service in conjunction with blog post searching and some are closely integrated with feed reading services.

Several characteristics of this task distinguish blog retrieval from typical ad hoc document retrieval. First, blog retrieval is a task of ranking document collections rather than single documents. In this respect, blog feed search bears some similarity to resource ranking in federated search. As a stream of individual entries, a blog feed can be viewed at multiple levels of granularity. We can represent each feed as a single large document for retrieval, or we can retrieve entries and aggregate an entry ranking into a feed ranking. Additionally, the set of entries for a blog are likely to have some topical relationship with each other and with the blog as a whole. If we choose to treat entries as individual documents, it may be possible to take advantage of the topical relationship between entries in our feed ranking.

We present a series of probabilistic retrieval models for blog retrieval. Through these models, we investigate the relationship between the topicality of individual entries and the blog as a whole, and we investigate the appropriate unit of representation for this task – whether it is the entry or the feed. These models extend a state-of-the-art approach previously developed for federated search to blog retrieval, the ReDDE algorithm proposed by Si and Callan [67]. Contrary to previous work in feed retrieval [5, 25, 66], we show that a federated search model with entries as the unit of retrieval can outperform a “large document” model that treats the whole feed as the unit of retrieval.

3.1.1 Probabilistic Retrieval Models for Feed Search

Previous research into feed search models has drawn analogies between this task and several other well-studied retrieval tasks: expert finding [9], cluster-based retrieval [65] and resource selection in distributed information retrieval [26, 66]. All of these tasks share the common goal of ranking collections of documents rather than single documents. Refer to Section 2 for a discussion of these other areas of research.

In the following sections we present a series of probabilistic retrieval models for feed search based on ones previously proposed for ad hoc retrieval and resource ranking in

10This section presents work originally published by Elsas, et al. [26].

11In this work, we refer to blogs (the collection of HTML web pages) and feeds (the XML syndication format version of the blog) interchangeably as there is a one-to-one correspondence between the two. Likewise, we refer to a blog post or permalink document (the HTML page) and a feed entry (an XML element within a feed) interchangeably.
federated search. Throughout this section, we follow the variable naming conventions in Table 1.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q, E, F, C$</td>
<td>Query, Entry, Feed, Collection</td>
</tr>
<tr>
<td>$q_i, \psi_i$</td>
<td>Query terms, query features</td>
</tr>
<tr>
<td>$tf_{i; M}$</td>
<td>Frequency of term (or feature) $t_i$ in document $M$</td>
</tr>
<tr>
<td>$</td>
<td>M</td>
</tr>
<tr>
<td>$N_C$</td>
<td>Number of documents in collection $C$</td>
</tr>
<tr>
<td>$N_F$</td>
<td>Number of entries in feed $F$</td>
</tr>
</tbody>
</table>

Table 1: Variable naming conventions.

### 3.1.2 Large Document Model

The first and simplest model treats each feed as a single monolithic document, ignoring any distinction between individual entries within those feeds. This baseline model is a simplified version of the unexpanded large document model from [5, 25], the best performing retrieval model without query expansion in the TREC 2007 Feed Distillation task. Keeping the same naming convention, we refer to this model as the large document model.

In comparison to previous work on resource ranking in federated search, this model is similar in spirit to the CORI algorithm, which creates pseudo-documents for each collection using collection term frequency statistics [16, 18]. Pseudo-documents are then ranked by their similarity to the query. Our large document model uses a similar approach, representing each feed by a concatenation of all its entries. We derive the large document model as follows, ranking feeds by their posterior probability given the query

$$P_{LD}(F|Q) = P_{LD}(Q, F) / P(Q)$$

$$\text{rank} = P(F) P_{LD}(Q|F)$$

where the query likelihood component is estimated with Dirichlet-smoothed maximum likelihood estimates [80]

$$P_{LD}(Q|F) = \prod_{\psi_i \in \Psi(Q)} P_{LD}(\psi_i|F)^{w_i}$$

$$= \prod_{\psi_i \in \Psi(Q)} \left( \frac{tf_{\psi_i; F} + \mu P_{MLE}(\psi|C)}{|F| + \mu} \right)^{w_i}.$$  \hfill (8)

The $\psi_i \in \Psi(Q)$ are query features as used in Metzler’s full dependence model[53] (query term unigrams and term windows) and $\mu$ is a smoothing parameter estimated from training data. Weights on the query features, $w_i$, are taken directly from previous work and have been shown to perform well across a variety of tasks and collections [49, 50].

Our implementation of this retrieval model is described with the following Indri\textsuperscript{12} query

\textsuperscript{12}http://www.lemurproject.org/indri/
template

```ruby
#combine(  #prior(prior name)
  #weight(0.8  #combine(unigram query)
    0.1  #combine(ordered window query)
    0.1  #combine(unordered window query))
),
```
a combination of a document prior and a dependence model query [53].

The feed prior component, shared between this model and the small document models introduced below, is used to incorporate query independent features into the ranking algorithm. See Section 3.1.3 for a detailed explanation of feed priors and how they are used in our models.

### 3.1.3 Small Document Models

The next set of models treat blog feeds as collections of individual documents — the blog’s constituent entries. Retrieving information sources as collections rather than single entities has been an effective approach in federated search, cluster based retrieval and expert finding. Additionally, decomposing our retrieval task in this way enables us to model the relationship among entries or between the entry and the feed, measuring how “central” the entry’s language is to that of the entire feed.

Keeping these concerns in mind, our small document model is derived as follows, again ranking feeds by the posterior probability of observing the feed given the query

\[
P_{SD}(F|Q) = \frac{1}{P(Q)} \sum_{E \in F} P_{SD}(Q, E, F)
\]

\[
= \frac{P(F)}{\sum_{E \in F} P(Q|E, F)P(E|F)}
\]

\[
= \frac{P(F)}{\sum_{E \in F} P(Q|E)P(E|F)}
\]

where the last line holds if we assume queries are conditionally independent of feeds given the entry. This model scores feeds as a weighted average of their constituent entry scores.

The entry weighting in this model is defined by the centrality component, described in more detail below.

The model above extends the one proposed in [5, 25], which is loosely based on the ReDDE federated search algorithm [67]. ReDDE is a resource ranking algorithm which scores a document collection, \(C_j\), by the estimated number of relevant documents in that collection

\[
Rel_q(C_j) = \sum_{d_i \in C_j} P(rel|d_i)P(d_i|C_j)N_{C_j},
\]

where \(N_{C_j}\) is an estimate of total number of documents in collection \(C_j\). The ReDDE model favors large collections, a desirable property when ranking by the expected number of relevant documents. But in our task, high traffic blogs may not necessarily be more relevant than infrequently updated blogs. The ReDDE analog of our centrality component,
$P(d_i|C_j)$, is uniform on a per-collection basis. We extend this to a true measure of centrality rather than simply a means to balance collections of different sampled sizes.

**Query Likelihood**

The query likelihood component of our small document model is estimated similarly to the large document model, using the same full dependence model query features. For the small document model, we use Jelinek-Mercer smoothing \[80\] rather than Dirichlet (Equation 8), enabling us to combine evidence from the entry, feed and collection

$$P_{JM}(Q|E) = \prod_{\psi_i \in \Psi(Q)} P_{JM}(\psi_i|E)^{w_i}$$

$$= \prod_{\psi_i \in \Psi(Q)} (\lambda_E P_{MLE}(\psi_i|E) + \lambda_F P_{MLE}(\psi_i|F) + \lambda_C P_{MLE}(\psi_i|C))^{w_i} \quad (10)$$

where $\sum \lambda_s = 1, \lambda_s \geq 0$ and $P_{MLE}(\psi_i|M) = \frac{t_{\psi_i|M}}{|M|}$. Again, the smoothing parameters $\lambda_s$ are estimated from training data. Although the small document model cannot be completely expressed in the Indri query language, the query likelihood scoring is identical to a dependence model query, retrieving entries rather than feeds.

**Entry Centrality**

The entry centrality component of our model serves two purposes. First, because we want to favor relevant entries that are also representative of the entire feed, the centrality component measures how closely the language used in the entry’s text resembles the language of the feed as a whole. This has the effect of down-weighting the influence of an outlier entry that happens to be relevant to the query.

The second purpose of the $P(E|F)$ component is to balance the scoring across feeds with varying numbers of entries. Without this balancing, the summation in the small document model, Equation 9, would favor longer feeds.

Our entity centrality component is proportional to some measure of similarity between the entry and the feed, $\phi$, normalized to be a probability distribution over all the entries belonging to this feed

$$P(E|F) = \frac{\phi(E,F)}{\sum_{E_i \in F} \phi(E_i,F)} \quad (11).$$

In general, any measure of similarity could be used here, for example, K-L divergence or cosine similarity. In our experiments we evaluated two centrality scoring functions. As a means to assess the effect of the centrality component of our model, our first scoring function is uniform, i.e. no centrality computation

$$\phi_{CONST}(E,F) = 1.0$$

and the centrality component of our model using this scoring function only serves to normalize for feed size. The second scoring function computes a centrality measure based on the geometric mean of term generation probabilities, weighted by their likelihood in the
entry language model

\[
\phi_{GM}(E, F) = \prod_{t_i \in E} P(t_i|F)^{P(t_i|E)} = \left( \prod_{t_i \in E} P(t_i|F)^{t_i|E} \right)
\]  (12)

where we estimate the feed language model as follows, again taking care to control for varying entry lengths

\[
P(t_i|F) = \frac{1}{N_F} \sum_{E_j \in F; j=1}^{N_F} P_{MLE}(t_i|E_j).
\]

This scoring function is similar to the un-normalized entry generation likelihood from the feed language model.

In our implementation, the product in Equation 12 is only performed over the query terms, thereby providing a topic-conditioned centrality measure biased towards the query. Additionally, significant efficiency improvements can be realized by only taking the product over the query terms rather than the entire entry vocabulary.

In some sense, the entry centrality term in our model is similar to Hannah et. al.’s blog cohesiveness measure [30]. However, our centrality measure is more appealing in several ways: (1) it has a direct probabilistic interpretation in the model, (2) it gives an entry-specific score instead of a global feed score, and (3) as described above, this score can be conditioned on the query, providing a query-specific centrality measure.

The formulation of our centrality measure, Equation 11, has the tendency to inflate the scores of entries belonging to shorter feeds. Smoothing the centrality normalization could be one way to control for this, for example with “field smoothing” as proposed by Zhao and Callan [81], which extends the Dirichlet smoothing used in Equation 8 to structured documents. In this work we chose to use the feed prior as a means to favor feeds based on their size, thereby separating the centrality and feed size components of our feed ranking model.

Feed Prior

The feed prior component, \(P(F)\), provides a way to integrate query-independent factors into the feed ranking. In this work we use the feed prior to favor longer feeds, which without any knowledge of the query are more likely to contain relevant entries. This also has the effect of controlling for the overly-optimistic centrality scoring for short feeds.

We evaluate two feed priors in this work: one which grows logarithmically with the feed size, \(P_{LOG}(F) \propto \log(N_F)\), and a uniform feed prior that does not influence the document ranking at all, \(P_{UNIF}(F) \propto 1.0\). Note that our small document is very similar to the ReDDE model if we use the constant entry centrality measure, \(\phi_{CONST}\), and choose a prior that grows linearly with the size of the feed, \(P_{LIN}(F) \propto N_F\). Initial testing with a linear prior for this task, however, yielded degraded performance.

3.1.4 Retrieval Model Experiments

We evaluated these models using the 45 topics and relevance judgements from the 2007 TREC Feed Distillation task on the BLOG06 test collection [45, 47], using only the topic
As stated above, this task is ranking blog feeds in response to a query, not blog posts. BLOG06 is a collection of blog home pages, blog entry pages (permalink links) and XML feed documents. For these tests, we chose to index only the feed XML documents. Although these documents potentially contain partial content of the blog posts rather than the full text, they tend to be less noisy. The feed documents typically do not contain advertisements, formatting markup or reader comments, all of which could lead to degraded retrieval performance. We index the feeds as structured documents containing a series of \texttt{<entry>} elements for each feed entry, allowing index reuse across experiments.

All results reported are from 5-fold cross validation to choose the smoothing parameters used in the query likelihood calculations described above (Equations 8 and 10), and all experiments were performed with an extended version of the Indri search engine.

Our evaluations focused on the following questions: (1) does a small document retrieval model that attempts to control for varying entry length outperform the large document retrieval model that treats the feed as a single bag-of-words? (2) does a measure of entry centrality further improve performance? and (3) what is the effect of feed length?

<table>
<thead>
<tr>
<th>Model</th>
<th>Prior</th>
<th>Centrality</th>
<th>MAP</th>
<th>P@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD</td>
<td>(P_{UNIF})</td>
<td>-</td>
<td>0.290</td>
<td>0.400</td>
</tr>
<tr>
<td>SD</td>
<td>(P_{UNIF})</td>
<td>(\phi_{CONST})</td>
<td>0.277</td>
<td>0.391</td>
</tr>
<tr>
<td>SD</td>
<td>(P_{UNIF})</td>
<td>(\phi_{GM})</td>
<td>0.290</td>
<td>0.409†</td>
</tr>
<tr>
<td>LD</td>
<td>(P_{LOG})</td>
<td>-</td>
<td>0.188</td>
<td>0.320</td>
</tr>
<tr>
<td>SD</td>
<td>(P_{LOG})</td>
<td>(\phi_{CONST})</td>
<td>0.298⁺</td>
<td>0.418⁺</td>
</tr>
<tr>
<td>SD</td>
<td>(P_{LOG})</td>
<td>(\phi_{GM})</td>
<td>0.315†⁺⁺</td>
<td>0.424⁺⁺</td>
</tr>
</tbody>
</table>

Table 2: Mean Average Precision and Precision at 10 for the large document (LD) and small document (SD) retrieval models with different centrality measures and different feed priors (Section 3.1.3). Statistical significance at the 0.05 level is indicated by † for improvement from \(\phi_{GM}\), + for improvement from \(P_{LOG}\) and * for improvements over the best LD model.

The full set of results is presented in Table 2 with significance testing performed with the Wilcoxon Matched-Pairs Signed-Ranks Test. First, looking at the top three rows using the uniform feed prior \(P_{UNIF}\), we can see that the large document and small document retrieval models perform comparably when using the centrality measure \(\phi_{GM}\), but without the centrality measure \(\phi_{CONST}\) the large document model outperforms the small document model. Next, when using the logarithmic feed prior \(P_{LOG}\), the small document model clearly outperforms the large document model. The best small document model performance \((\phi_{GM} \text{ and } P_{LOG})\) significantly outperforms the best large document model \((P_{UNIF})\) and using the centrality measure \(\phi_{GM}\) clearly helps the small document model performance across tests. The feed prior has the opposite effect on the small and large document models, significantly hurting performance on the large document model and helping on the small document model. This indicates that the benefit of this prior term may come from the interaction between the prior and the centrality components of the small document model, not from an intrinsic property of large feeds being more relevant. Further evaluation of these models is necessary to fully understand this interaction.
3.1.5 Conclusion

In this section, we have presented our experiments on blog feed search. We developed several probabilistic feed retrieval models, showing that existing federated search algorithms can be effectively adapted to this task. The best performing federated small-document model showed significant improvement over a strong large document model – the best non-expanded submission at the 2007 TREC Blog Distillation task – yielding a 9% improvement in MAP and an 6% improvement in P@10. This result is contrary to those previously published by us and others [5, 9, 25, 65] and demonstrates the need to effectively model the topical relationship between the feed and its entries. The major contribution of the small document model presented here is that it provides a novel and principled mechanism to measure the topical relatedness of the document to its collection and to integrate that into the retrieval algorithm.

The retrieval models presented here are not specific to blog feed retrieval and may have applications beyond this task. The small document model presented here can be sensibly applied to any retrieval problem where collections of topically related documents are ranked, including email or newsgroup thread retrieval, web results collapsing, cluster-based retrieval, and other federated search tasks. As part of the proposed work, we will extend these retrieval models and apply them to tasks beyond blog feed search.

3.2 Online Forums Search

Online forums, or message boards, contain a wealth of user generated content over a wide range of topics: from computer hardware to movies reviews and commentary to specific health issues. The contributors to online forums are often domain experts and these social information spaces host in some cases many millions of archived messages. Access to this historical information, however, is often rudimentary, providing the most basic of browsing interfaces and simple keyword message-searching facilities. This is evident in the content of several messages posted to one message board, the MacRumors forum (see Section 3.2.2 for a description of this dataset):

There are a ‘few’ threads already on similar subjects (did you search?)

— thread 140098

i searched the forums, but couldn’t find anything related, so i googled, and what’s the first hit? a macrumors form topic.

— thread 192892

Did you bother searching for one of the other million threads like this?

— thread 412491

These quotes reflect not only a lack of familiarity with the search features on the forum, but also a lack of effectiveness of those features.

13This section describes work currently under review [24] as well as ongoing work in progress.
In this section we explore the problem of information access in archives of online forum data. We describe the structure in online forums, and detail the construction of an information retrieval test collection for use in studying search tasks in online forums. Several baseline retrieval models for online forum search are then presented, as well as preliminary experimental results using this dataset.

### 3.2.1 The Structure of Online Forums

Online forums are highly structured, often with different levels of organizational granularity and different axes of organization. Messages are typically grouped into *message threads*, representing a single conversation between a group of contributors. A message thread has a single *start message* contributed by the *thread starter* and zero or more *response messages* contributed by *respondents*. Message threads are frequently displayed chronologically in a “flat” structure where each message in the thread has at most one response. Some online forums support hierarchical organization, where each message may have more than one response. Figure 1 shows this thread-level organization.

![Figure 1: Online Forum Message Thread Organization, showing a simple hierarchical message organization.](image)

Message threads are often grouped into *sub-forums*, which usually represent sub-topics discussed in the online forum. Sub-forums can, in turn, be organized hierarchically, with child sub-forums representing more specific topics. The thread starter selects a sub-forum when composing the start message, and thus these sub-forums can be thought of as an author assigned thread classification. In the case of the MacRumors forum (discussed below) these sub-forums are focused on specific hardware or software products. Figure 2 shows this subforum-level organization.

Another axis of organization is time. Messages in the online forum are posted at a specific time, which is recorded in the message metadata. Online forum interfaces frequently order threads by the date of their most recent contribution, allowing users to browse conversations recently active. As can be seen in Figure 5, some threads are active for a very short time, while others can have contributions to the thread over the course of several years.
From an information retrieval perspective, the rich organization present in online forums provides several interesting challenges. The first challenge is that there are several possible units of retrieval. Depending on the task, an information seeker may be interested in retrieving individual messages, entire message threads, or even an entire sub-forum. We may also envision a situation where, like the expert finding tasks described in Section 2.3.3, information seekers are interested in finding authors with a high level of expertise on a particular subject. The unit of retrieval is primarily dictated by the task, and in this work we focus on retrieving message threads. We believe this to be a generally useful unit of retrieval, and further discussion of the retrieval granularity is given below.

A second challenge in information retrieval over this type of collection is that of leveraging this structure within ranking algorithms. In the task of message thread ranking, an information seeker may be more interested in threads residing in a particular sub-forum that is closely related to their query. Or, users may be interested in threads with significant contributions from authors who have a high level of expertise with respect to their information need. To leverage this type of structure in ranking, we must develop algorithms that can model the expertise of authors and the affinity of the query to the different sub-forums.

### 3.2.2 Dataset Description

The work presented here uses a recent crawl of a technically-oriented, online message board, the MacRumors Forum\(^{14}\). This is a large online forum dedicated to the discussion of news and opinion relating to the computer manufacturer, Apple, Inc. The crawl was conducted in March of 2008 and contains over 3 million posts. Detailed dataset statistics are given in Table 3. Figure 3 shows a distribution of thread lengths. Figure 4 shows evolutionary statistics of the forum over time.

\(^{14}\)http://forums.macrumors.com
To create the dataset, each HTML page corresponding to a single thread was downloaded. Thread are numbered sequentially and this number is part of the thread URL (e.g., forum.com/showthread.php?t=123). Through this simple programmatic access of forum threads, it is possible to make an exhaustive crawl of all threads on the forum.

The thread HTML pages downloaded in this manner contain the contents of the start message and the first twenty-four response messages in the thread. These messages are extracted from the HTML, retaining structural information such as thread titles, message titles, user IDs and message time stamps. Note that although an exhaustive thread crawl was performed, not all forum messages were extracted. Because only the first 25 messages for each thread were downloaded, 9.8% of the threads in the crawled collection are incomplete. Although this is a relatively small fraction of incomplete threads, these missing messages do account for a sizable portion of the entire forum: roughly 30% of the total message on the forum at the time of the crawl are not included in our dataset.

Towards an Information Retrieval Test Collection

As stated above, these online forums host an enormous archive of historical information, over 3 million messages in the case of the MacRumors forum. Search-based access to this
Figure 4: MacRumors.com forum characteristics over time. From top to bottom: Daily message volume, with significant events indicated with colored triangles; Daily thread volume; Monthly user volume. Note: Drop in early 2008 is an artifact of the data collection method.

information, however, is often rudimentary. Because of the difficulty in accessing the forum archives, users often post questions to message boards that may have been answered in previous threads in the archive. Commonly, when this happens, another user responds to that question with a link to a previous discussion possibly containing the answer. We can leverage this interaction between online forum users to build an information retrieval test collection. The original question can be considered a query and the linked-to thread a relevant document.

Identifying Information Needs and Relevant Documents

We make the assumption that the typical useful unit of retrieval for message board search is the message thread. Although this is certainly not always the case — sometimes a single message may fully answer an information need — the thread provides useful conversation context and discussion. When viewing the message board, a thread-view is typically most convenient. Additionally, of all the intra-forum linking, 86% of the links refer to other threads, rather than posts or other possible units of retrieval. Table 4 shows the volume of linking to different possible units of retrieval in the MacRumors forum.

To build a thread-retrieval test collection, we isolated all response messages that contained a link to other threads in the same collection. These candidate “answer messages” may be an instance of a forum user answering a start message question with a link to a previously posted relevant thread. Over 17,000 candidate answer messages were identified, and a random sampling of 550 of the corresponding threads were manually annotated for containing a question/link-answer pair. A thread contains a question-answer pair when
the following conditions hold true:

1. A response message provides a hyperlink to a previous thread in the message board.

2. The start message in this thread contains a question that is answered by the linked-to thread.

3. Subsequent response messages in the thread do not indicate the linked-to thread is irrelevant to the original question.

Many instances of within-forum linking are common, and do not necessarily indicate a question/answer interaction. Often links indicated a forum user was experiencing similar problems as the thread starter (but providing no answers), links to a news topic that may have been previously discussed, or links advertising items for sale by forum users. Because of the volume and diversity in link types in the forum data, less than 10% of the sampled links indicated a forum thread that directly answered the thread starter’s question. In total, 48 question-answer pairs were identified in this test collection.

Previous work has shown that high quality answers (and questions) in online social media tend to be longer than lower quality content [1]. It follows that relevant threads, which presumably contain a high-quality answer to the question in the start message, may be longer than the typical forum message. This hypothesis is borne out in the data, with the mean message size in answer threads being 93.7 tokens as compared to 81.8 tokens in the entire collection (see Table 3). Additionally, answer threads tend to be longer, containing on average 43.5 messages compared to the average of 12.4 messages per thread in the collection. Both of these differences are statistically significant with a 1-tailed t-test ($p < 0.001$ and $p < 0.005$ respectively).

Figure 5 shows the temporal distribution of the question message (black dots) and the answer thread activity (black lines). While many questions messages are composed shortly after the answer thread is started, this is not necessarily always the case. Some answer threads are active for several years before the question is posted. The answer threads are also active for a wide variety of time, reflecting the overall popularity of those threads. Some of the more popular answer threads have many messages that are contributed over the course of two or more years. Others are very short lived, where messages may only be contributed on a single day.

This type of test collection creation does have some distinct advantages over other typical retrieval test collections. First, the queries represent real information needs of real users of the online forum. These information needs are also much more verbose than typical keyword queries on a web search engine, providing a retrieval system more evidence with

<table>
<thead>
<tr>
<th>Link Target</th>
<th>Volume</th>
<th>Percent of Total Link Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thread</td>
<td>27,068</td>
<td>86.43%</td>
</tr>
<tr>
<td>User profile page</td>
<td>2,071</td>
<td>6.61%</td>
</tr>
<tr>
<td>Message</td>
<td>1,140</td>
<td>3.64%</td>
</tr>
<tr>
<td>Subforum</td>
<td>688</td>
<td>2.20%</td>
</tr>
<tr>
<td>Search result page</td>
<td>350</td>
<td>1.12%</td>
</tr>
</tbody>
</table>

Table 4: Volume of intra-forum linking in the MacRumors Forum.
which to use in relevance scoring. The “relevance judgement”, provided by another forum user linking to a previous thread, also presents in-situ relevance information — sensitive not only to the original question, but also to the overall nature of the forum and the time when the question was asked.

There are several drawbacks inherent in this type of corpus creation, most importantly with regard to the exhaustiveness of the relevance assessment. Typically in TREC-style collection development [71], ranked results from several retrieval systems are pooled and those pooled documents are assessed for relevance. When the systems’ output is sufficiently diverse and relevance assessment is sufficiently deep, this produces a reasonably complete relevance assessment for each query — if a relevant document is in the collection, it would most likely be retrieved by one of the systems and be judged by being admitted into the pool. The method of collecting relevance judgements presented here, on the other hand, will not produce anything close to an exhaustive set of relevant threads. In the great majority of cases, only a single thread is linked to in a subsequent reply message. There is no guarantee that this thread is the best or only relevant thread in the collection. For this reason, we must take care in evaluating thread retrieval algorithms with this dataset, and not to assume non-judged threads are necessarily irrelevant.

One additional facet of this test collection is significantly different than IR test collections typically used. The online forum data is temporal, with threads and messages constantly being added to the collection. For this reason, any evaluation with this data must take into account the time at which a question is asked, and only consider messages created prior to that time for evaluation.

### Generating Queries

Queries were created from the original question posts in several ways. First, the text of the original post was extracted from the post body and title, eliminating any HTML, quoted text or “signature” text. Several methods of generating queries from these messages...
were explored:

1. **FULL**: All the text from the body and title of the original message

2. **FULL.t**: The text only from the title of the original message

3. **CLEAN**: A manually “cleaned” version of the original message text, eliminating any text unrelated to the central question of the message. This includes introduction comments by the poster, phrases like “Hello”, “Thank You” or “Any suggestions”, and general commentary not related to the topic of the question

4. **CLEAN.t**: Processed similarly to **CLEAN**, but only text from the title of the message

Preliminary testing indicated that the **FULL** and **FULL.t** query sets were too noisy to yield reasonable retrieval results, while the **CLEAN.t** query set lacked critical content expressed in the message body. For this reason, all tests reported below use the **CLEAN** query set.

### 3.2.3 Retrieval Models for Forum Search

In this section we describe a variety of retrieval models that can be applied to thread search in online forums. The baseline models described here are taken from previous work in blog feed search [26], resource selection [65] and expert finding [46]. These models are described in more detail in Sections 3.1 and 2. These initial models focus only on the structure of the message thread, and ignore any other structure in the collection, such as authorship or subforum information.

**Baseline Thread Ranking Models**

We can think of the task of thread ranking as similar to blog feed search described in the previous section. In this case, a thread is analogous to a blog feed and a message is analogous to a blog post. When ranking threads, we have a similar goal in mind — to identify threads with a “central” focus on the topic described by the query. To this end, we apply blog search models directly to the task of thread ranking. The analogue of the large- and small-document models used for blog feed search (Equations 7 and 9) are given below:

\[
P_{LD}(T|Q)_{\text{rank}} = \frac{P(T)}{\text{Thread Prior}} \cdot \frac{P(Q|T)}{\text{Query Likelihood}}
\]

\[
P_{SD}(T|Q)_{\text{rank}} = \frac{P(T)}{\text{Thread Prior}} \sum_{M \in T} \frac{P(Q|M)}{\text{Query Likelihood}} \cdot \frac{P(M|T)}{\text{Message Centrality}}
\]

where in this case we are ranking thread \(T\) with respect to a user’s query \(Q\) rather than blog feeds. As before, we can view the large document model as representing threads as a concatenation of their respective messages and estimating probabilities based on that “large” document. The small document model, on the other hand, assigns scores to messages \(M\) and then aggregates those scores into a thread score via a weighted average.

As with blog feed search, we may be interested differentially weighting individual messages in a thread, for example favoring messages that more closely resemble the “central”
topic of the thread. We employ a similar technique to the task of thread ranking, inte-
grating a measure of the affinity of a message to the thread via the message centrality
component of the small document model. In this work, as before, we investigate both a
uniform message centrality measure, as well as a centrality measure based on the generation
probability of the message given the thread. See Section 3.1.3 for a detailed description of
the $\phi_{\text{CONST}}$ and $\phi_{\text{GM}}$ centrality measures which are also used here.

These retrieval models view all the messages in the thread as important for thread
ranking, although possibly re-weighted with a centrality measure. For this reason, we refer
to these models as inclusive ranking models. This is in contrast to the selective ranking
models described below.

### Selective Thread Ranking Models

In addition to the large- and small-document models, we also apply several thread re-
trieval models that rank threads by selective scoring of their constituent messages. In these
models, we make the assumption that the every message in the thread is not necessarily
useful for thread ranking, and by discarding some messages we can improve the overall
retrieval performance.

The first of the selective ranking model is based on the cluster selection model proposed
by Liu and Croft [41] and applied to blog feed search by Seo and Croft [65]. This model,
referred to as Pseudo-Cluster Selection (PCS), is given by the following:

$$P_{\text{PCS}}(Q|T) = \left( \prod_{1 \leq i \leq k} P(Q|M_i) \right)^{\frac{1}{k}}$$  \hspace{1cm} (15)

where the $i$ indexes the top-scoring $M_i$ messages in the thread with respect to the query
$Q$. Rather than combining message scores via a arithmetic mean as in the small document
model (Equation 14), this model first selects the top-$k$ messages and then combines their
scores via a geometric mean. For the experiments presented here, we let $k = 5$ which has
shown good performance in other tasks such as blog feed search.

Two additional selective retrieval models are also applied to this task. These two models
score a thread based on a single message in that thread, either the start message or the
single highest scoring message in the thread.

$$P_{\text{ST}}(Q|T) = P(Q|M_{\text{START}})$$ \hspace{1cm} (16)
$$P_{\text{MAX}}(Q|T) = \max_{M_i \in T} P(Q|M_i)$$ \hspace{1cm} (17)

where $M_{\text{START}}$ is the thread’s start message. Note that the $P_{\text{MAX}}$ ranking model is
equivalent to the pseudo-cluster selection model with $k = 1$.

### 3.2.4 Evaluation, Results & Analysis

In this section we present initial experimental results using the baseline retrieval models for
message thread search. As stated previously, due to the lack of coverage of our relevance
assessment in this test collection, we must take care not to assume un-judged documents are
not relevant. For this reason, a recall-oriented evaluation is appropriate and our primary
evaluation measures are recall at various cutoff levels (R@10, 20, 30, 100).
These measures are particularly coarse and may not provide much distinction between ranking algorithms. For most queries, we have only a single relevant document. We also report Mean Reciprocal Rank (MRR), which is an evaluation measure typically used when there is a single known-relevant item in the collection.

Complete performance results are given in Table 5. In this table we can see several clear trends. First, comparing at the large document (LD) and small document models (SD, MAX, ST and PCS), the small document models consistently outperform the large documents. Second, the selective methods of thread ranking (MAX, ST and PCS), which discard information in some of the messages when scoring the thread, outperform the methods that use all of the messages scores (SD). This indicates that when ranking threads in online forums, not all messages are useful in judging the relevance of a thread, and frequently the score of only a single message (as in MAX) achieves the best performance. In all cases except R@100, the best performing selective model significantly outperforms the LD and SD + φGM retrieval models using a 2-tailed paired t-test. Among those selective methods, there is not a clear winner.

<table>
<thead>
<tr>
<th>Retrieval Model</th>
<th>MRR</th>
<th>Recall at 10</th>
<th>Recall at 20</th>
<th>Recall at 30</th>
<th>Recall at 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD</td>
<td>0.0928</td>
<td>0.1553</td>
<td>0.2436</td>
<td>0.3113</td>
<td>0.5598</td>
</tr>
<tr>
<td>SD + φUNIF</td>
<td>0.0987</td>
<td>0.2283</td>
<td>0.3481</td>
<td>0.4379</td>
<td>0.6448</td>
</tr>
<tr>
<td>SD + φGM</td>
<td>0.0922</td>
<td>0.1867</td>
<td>0.2993</td>
<td>0.4170</td>
<td>0.6239</td>
</tr>
<tr>
<td>MAX</td>
<td>0.1491</td>
<td>0.3095</td>
<td>0.3920</td>
<td>0.4587</td>
<td>0.5984</td>
</tr>
<tr>
<td>ST</td>
<td>0.1570</td>
<td>0.3253</td>
<td><strong>0.4675</strong></td>
<td><strong>0.5561</strong></td>
<td>0.6180</td>
</tr>
<tr>
<td>PCS</td>
<td><strong>0.1902</strong></td>
<td><strong>0.3308</strong></td>
<td>0.4031</td>
<td>0.4158</td>
<td><strong>0.6491</strong></td>
</tr>
</tbody>
</table>

Table 5: Complete performance results, online forum thread ranking. Large Document Model (LD), Small Document Models with different centrality measures (SD), Max-Message (MAX), Start-Message (ST) and Pseudo-Cluster Selection (PCS) evaluated at Mean Reciprocal Rank (MRR) and Recall at various cutoffs.

Hidden-Success Analysis & Limitations of the Current Test Collection

As stated previously, due to the limited relevance information in this test collection, we focus on recall-oriented evaluation measures. The absolute values of these performance metrics, however, indicates that even the best-performing retrieval algorithm may be performing unsatisfactorily. Most queries have only one relevant document, and a R@10 value of 0.37 indicates that 30 of the queries, more than half, failed to find a relevant thread in the top 10 results. A R@100 value of 0.68 indicates that approximately 15 queries failed to retrieve any relevant results in the top 100 documents.

We manually inspected the top 10 results of those failed queries. Through this, we identified more than 15 likely relevant results that were retrieved in the top 10. This is an indication that the relevance information in our test collection may be insufficient to adequately evaluate the thread ranking algorithms, particularly in a ranking task such as this when high-precision is likely to be an import factor.
3.2.5 Conclusion

In this section we discussed the task of search in online forums. We developed an information retrieval test collection for this task with data collected from a large and active online community. To build this test collection, we presented a novel approach to identifying information needs and relevant threads within the collection by leveraging the linking behavior of members of this online community.

Using this test collection, we evaluated several baseline thread retrieval models. These models were adapted from other tasks such as blog feed search, expert finding and cluster-based retrieval. We found that selective ranking algorithms — algorithms that discard information from some of the messages in the thread — outperform methods that use all of the messages in the thread. Although these results are preliminary and the test collection may need to be refined, we believe that these results show promise.

In following sections will will present several directions for both building a more robust and useful online forum test collection, as well as methods of leveraging the rich structure of online forums in ranking algorithms.
4 Planned Thesis Work

The primary focus of this work is the development and evaluation of search algorithms that leverage the collection structure present in conversational social media archives. We plan to investigate three ways to leverage specific types of structure, with the goal of significantly improving retrieval performance. We will investigate:

1. Modeling **topical relationships** among elements when ranking sub-collections. This will be done through measures like *centrality* (described in Section 3.1.3) as well as other measures to be developed through the course of this thesis work.

2. Modeling **author expertise** when ranking documents in collections with multiple contributions from many authors.

3. Using the collection’s **topical organization** when ranking documents in collections with broad document classification schemes.

In addition to evaluating retrieval algorithms that leverage this type of collection structure, we may also evaluate methods to leverage these structures outside of document ranking. For example, the online forum datasets described previously can be used for several classification tasks, such as response prediction (e.g. which authors will respond to a given message), link prediction (e.g. which threads will be linked-to from a given thread) or sub-forum classification (e.g. which sub-forum is assigned given a thread start message). Investigating these supplementary tasks could potentially shed some insight into which features are useful in general for modeling author expertise, for example.

The remainder of this section describes in detail the proposed work for this thesis, as well as these other areas of investigation that may be included in the final thesis. Table 6 shows a summary of the proposed thesis tasks, the current progress of the tasks, as well as dependencies among the tasks. Within each of the sections referred to in this table, we present a **proposed work** sub-section describing the detailed tasks to be completed. Each of these subsections identify the **required items** which are on the critical path of thesis completion and **optional items** which may be performed as time permits.

4.1 Retrieval Algorithm Development

The primary focus of this thesis work is to extend existing information retrieval algorithms to leverage collection structure in document ranking algorithms. In this section we propose two approaches to this task, both of which will be evaluated as part of the thesis work. The first approach is to extend the language modeling information retrieval framework to leverage collection structure in ranking. The second takes a learning-to-rank approach, and views this as primarily a problem of engineering features that encode the structural information.

The proposed work on retrieval algorithms described below will use thread search in online forums as an example search task, but these techniques are easily applicable to other conversational social media collections. For example, public email list archives provide similar structure, with email messages organized into threads, threads belonging to mailing lists, and contributions by many authors.
<table>
<thead>
<tr>
<th>Task</th>
<th>Subtask</th>
<th>Dependency</th>
<th>Status</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. Algorithm Development</td>
<td>I.i. Probabilistic Algorithm</td>
<td></td>
<td>in progress</td>
<td>4.1</td>
</tr>
<tr>
<td></td>
<td>I.ii. Feature Engineering, Machine Learning Algorithm</td>
<td></td>
<td>proposed</td>
<td></td>
</tr>
<tr>
<td>II. Collection Development</td>
<td>II.i. Obtain Document Collection</td>
<td></td>
<td>complete</td>
<td>4.2</td>
</tr>
<tr>
<td></td>
<td>II.ii. Identify Query Set</td>
<td>II.i.</td>
<td>in progress</td>
<td></td>
</tr>
<tr>
<td></td>
<td>II.iii. Create Document Pool</td>
<td>II.ii.</td>
<td>proposed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>II.iv. Relevance Assessment</td>
<td>II.iii.</td>
<td>proposed</td>
<td></td>
</tr>
<tr>
<td>III. Retrieval Tasks</td>
<td>III.i. Blog Feed Distillation</td>
<td>I.</td>
<td>complete</td>
<td>4.3</td>
</tr>
<tr>
<td></td>
<td>III.ii. Blog Post Search</td>
<td></td>
<td>proposed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>III.iii. Message Board, Thread Search</td>
<td>I., II.</td>
<td>in progress</td>
<td></td>
</tr>
<tr>
<td></td>
<td>III.iv. Email Message Search</td>
<td>I.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>III.v. Email Thread Search</td>
<td>I.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IV. Prediction Tasks*</td>
<td>IV.i. Response Prediction*</td>
<td>II.i.</td>
<td>proposed</td>
<td>4.4</td>
</tr>
<tr>
<td></td>
<td>IV.ii. Link Prediction*</td>
<td>II.i.</td>
<td>proposed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IV.iii. Sub-forum Classification*</td>
<td>II.i.</td>
<td>proposed</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Summary of proposed work and other areas of investigation. An asterisk (*) indicates areas of investigation that may be pursued over the course of the thesis work, but are not on the critical path for the thesis completion.

### 4.1.1 Collection Structure in the Probabilistic Retrieval Framework

There has been a large body of work in language modeling for retrieval of structured documents [13, 81], but very little work on extending the view of structured retrieval in the other direction — super-document rather than sub-document structure. In the context of blog feed search, we have already shown that modeling the topical relatedness of elements (the posts) to their collection (the blog) can improve retrieval performance. This is an example of leveraging only one type of collection structure in ranking, the relationship of individual element to its sub-collection.

In this section we present another example of how one might further extend the language modeling retrieval framework to leverage broader collection structure, using thread search in online forums as our motivating example. This retrieval model is a generalization of the small document model presented in section 3.1.3 and applied to both blog feed search and preliminary experiments in message thread search. The retrieval model presented here takes inspiration from a formal graphical model known as “mixtures of trees” which enables encoding of dynamic dependency structure as a hidden variable [48]. By mixing models with differing dependency structures, we can calibrate our ranking model to place more or less influence on different structural elements in the collection.

**Model Description**

To introduce this model, we will first consider a Bayesian network that represents
the relationships between all the elements of our online forum collection, the messages (M), threads (T), subforums (S) and authors (A). Taking a thread-centric view of these relationships, we can develop the following query-generation process:

1. Select a thread from our collection with some probability \( P(T) \).
2. From that thread, select an author and subforum with probabilities \( P(A|T) \) and \( P(S|T) \).
3. From the thread, author and subforum, select a message with probability \( P(M|T, A, S) \).
4. From that message language model, generate the query with probability \( P(Q|M) \).

This generative process corresponds to the graphical model depicted in Figure 6. \(^{15}\)

![Figure 6: Bayesian Network for thread ranking in online forum search.](image)

Using this independence structure, we can then rank threads by their likelihood given the query:

\[
P(T|Q)^{\text{rank}} = P(Q,T) = \sum_{M,A,S} P(Q,T,M,A,S) = \sum_{M,A,S} P(Q|M)P(M|S,T,A)P(S|T)P(A|T)P(T)
\]

Estimating the probabilities \( P(M|S,T,A) \) poses some challenges, and for this reason we employ the use of a unobserved choice variable \( Z \), which lets us model the influence of each conditioning variable separately. This variable can be thought of as describing how important each structural element in our collection (the thread, the subforum, or the author) are in addressing the information needs presented by the query. Conditioned on the value of \( Z \), we can then define three simplified networks (trees or chains) that encode only the dependence on each of these three elements. A graphical representation of this model is shown in Figure 7.

\(^{15}\)Step 2 in this generative process may seem initially counter-intuitive — it may make more sense to think of an author and subforum “generating” the thread rather than the opposite scheme presented here. This scheme makes more sense if we think of the subforum as equivalent to a class which is selected by the original author of the thread. Additionally, as we will see below, the structure presented here leads to some mathematical conveniences such as a model-wide thread prior \( P(T) \).
Using the independence structure described in this mixture of trees model (Figure 7), we can then re-write the thread likelihood ranking formula as follows, letting $P^i$ represent the probabilities with respect to each of our mixture components:

$$
P(T|Q) \overset{\text{rank}}{=} \lambda_T P^T(Q,T) + \lambda_S P^S(Q,T) + \lambda_A P^A(Q,T) \tag{21}
$$

$$
= \lambda_T \sum_M P^T(Q|M) P^T(M|T) P^T(T) + \lambda_S \sum_{M,S} P^S(Q|M) P^S(M|S) P^S(S|T) P^S(T) \tag{22}
$$

$$
+ \lambda_A \sum_{M,A} P^A(Q|M) P^A(M|A) P^A(A|T) P^A(T) \tag{23}
$$

Although this model allows for estimation of the probabilities in each of the mixture components differently, we make the assumption that $P^T(\cdot) = P^S(\cdot) = P^A(\cdot)$, yielding our final probabilistic thread ranking formula:

$$
P(T|Q) \overset{\text{rank}}{=} P(T) \left( \lambda_T \sum_M P(Q|M) P(M|T) \right) \text{ Thread Model} \tag{25}
$$

$$
+ \lambda_S \sum_{M,S} P(Q|M) P(M|S) P(S|T) \text{ Sub-forum Model} \tag{26}
$$

$$
+ \lambda_A \sum_{M,A} P(Q|M) P(M|A) P(A|T) \text{ Author Model} \tag{27}
$$

The mixture coefficients $\lambda_i \in [0,1], \sum_i \lambda_i = 1$ represent the probabilities that our unobserved choice variable reflects thread-, subforum- or author-influence, $P(Z = z_i) = \lambda_i$. Note that, if we assume $P(M|T) = 0$ for messages not contained within the thread and let
\( \lambda_S = \lambda_A = 0 \), then the mixture-of-trees model is identical to the small document model for message thread ranking, Equation 14.

This model has several appealing characteristics. First, it integrates subforum and author information into our thread ranking formula, and it does so similarly to previous models for expert search and cluster selection. For example, we may want to favor threads with strong contributions from authors who also have some expertise in topics related to the query. The author model portion of our ranking formula does exactly that. The first part, \( \sum_M P(Q|M)P(M|A) \), can be thought of as an expert search model (see Balog’s Model 2, Section 2.3.3), giving authors with a large amount of expertise with respect to the query a higher score. Combining this with the second part, \( P(A|T) \), assigns threads a high score if those expert authors have a strong contribution to the thread. An analogous argument can be made for the Sub-forum Model, which can be thought of as similar to a cluster- or resource-weighting step in our thread ranking.

Although this model does integrate sub-forum, author and thread information into the thread ranking, it ignores some of the interactions between these components. For example, an author’s expertise may be isolated to a particular sub-forum, and their contributions to other sub-forums could be considered less valuable. If that is the case, we may want to condition the authors’ expertise on the subform. In this way, the score assigned by the Author Model would vary depending on which subforum contains the message to be scored.

We leave formulations of probabilistic retrieval models that can capture these higher-order interactions between elements of the collection as a part of the proposed work.

This model is theoretically appealing, and provides a convenient and straightforward platform for integrating collection structure into the statistical language modeling information retrieval framework.

**Proposed Work**

The probabilistic thread retrieval model presented above provides a framework for incorporating collection-level structural evidence into the task of thread ranking. Several tasks remain in the refinement and further development of this model:

1. **Higher-order Interactions** (required): As stated above, one drawback to the model presented is its ignorance of higher-order interactions between collection elements, such as conditioning the Author Model score on the subforum. As part of the proposed work we must establish (1) how those interactions can be integrated into this or other probabilistic ranking models, and (2) if modeling those interactions is beneficial for thread search or other related tasks. This will require exploratory analysis of the data, identify whether and how authors’ contributions vary across subforums.

2. **Estimation** (required): Each component of the ranking model (e.g., \( P(Q|M) \), \( P(M|A) \), etc.) must be estimated from the collection statistics. Language modeling provides a useful framework for estimating the query generation probabilities \( (P(Q|M)) \), but lend no insight into the estimation of other probabilities in the model. As part of the proposed work, we must investigate methods of estimating these other model components. We have completed some of this work through in the context of blog feed search [5, 26], however more investigation is necessary, particularly in the application of this model to other domains.
3. **Parameter Fitting** (required): The mixture weights \( \lambda_i \) as well as smoothing parameters used throughout the model must be fit from training data. There is a large body of work establishing reasonable ranges of smoothing parameter values [49, 50, 58, 81] as well as applying machine learning techniques to learn parameters automatically [51, 52]. These machine learning techniques are particularly appealing — as retrieval models increase in complexity, the number of parameters increases, and brute-force techniques for maximizing performance over the parameter space are no longer feasible.

As part of the proposed work, we must identify and apply methods for parameter fitting in these retrieval models. By evaluating the effect of different parameters, such as the mixing weights \( \lambda_i \), to the overall algorithm performance, we can understand the importance of modeling author expertise or topical organization in the ranking tasks. This analysis is necessary to validate our original hypotheses.

4.1.2 **Feature-Based Ranking Models and Collection Structure**

In addition to investigating how collection structure can be integrated into probabilistic ranking models such as the one above, we also plan on a similar investigation with feature-based machine learning models for information retrieval. Many algorithms have been proposed to automatically learn ranking functions that combine a large number (possibly several hundred) of query-document pair features. This problem is commonly referred to as the “learning-to-rank” problem. These features often include scores from baseline ranking algorithms, term count statistics, or query-independent features such as in-link count [40]. Linear models, such as RankSVM [32] and perceptron variants [27] have been applied to this task, as well as non-linear neural network models such as RankNet [14] and LambdaRank [15]. In fact, many probabilistic retrieval models can be optimized through learning-to-rank methods [51] or viewed as feature-based ranking models [52].

We hypothesize that many aspects of collection structure that may benefit document ranking can be instantiated as real-valued features to use in a learning-to-rank algorithm. In a simple case, consider the mixture of trees model presented in the previous section. The scores produced by the Thread, Sub-forum and Author models can be thought of as three feature, and the learning task is to optimize their mixing weights \( \lambda_S, \lambda_T \) and \( \lambda_A \). We may also want to expand this feature set with priors on authors, sub-forums and threads or other feature that may be difficult to integrate into the probabilistic ranking model described above.

**Ranking Features**

There are several ways to characterize features we may wish to investigate for learning-to-rank algorithms. For example, we may wish to consider features with respect to each of the object types in our collection, and how they relate to the unit of retrieval. In the online forum dataset when retrieving threads, we may create feature sets for threads, authors or sub-forums individually. Table 7 lists a variety of features that may be useful for thread ranking in online forums. The features in this table have been categorized into which objects in the collection (message, thread, subforum, author) are used to generate the feature value. By categorizing the features used in a ranking algorithm in this way, we can perform ablation studies to identify which aspects of our collection structure are
Probabilistic Objects

Feature Interpretation M T S A

<table>
<thead>
<tr>
<th>Query Dependent Features</th>
<th>Probabilistic Interpretation</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Relevance to Query</td>
<td>$P(Q</td>
<td>M)$</td>
</tr>
<tr>
<td>Sub-forum Relevance</td>
<td>$P(Q</td>
<td>M)P(M</td>
</tr>
<tr>
<td>Author Expertise</td>
<td>$P(Q</td>
<td>M)P(M</td>
</tr>
<tr>
<td>Subforum-Conditioned Author Expertise</td>
<td>$P(Q</td>
<td>M)P(M</td>
</tr>
</tbody>
</table>

| Query-Independent Features                                    |                             |         |
| Message Length                                                |                             | ●       |
| Message Grammaticality                                         |                             | ●       |
| Thread Length                                                  |                             | ●       |
| Subforum Size                                                  | $P(S)$                      | ●       |
| Author Post Volume                                             | $P(A)$                      | ●       |
| Subforum-Conditioned Author Post Volume                        | $P(A|S)$                    | ● ●     |

Table 7: Learning-to-Rank Features. Right columns indicate which collection elements are used in the feature generation: the message (M), thread (T), sub-forum (S) or author (A).

It can be easier to integrate some types of evidence in the feature-based ranking setting rather than the probabilistic model presented above. For example, we can generate a graph with the different element types in our collection as the vertices and edges representing the strength of association between those elements — an edge could be drawn between authors and the threads they contribute to, between a thread and the subforum it belongs to, etc. From this graph we can generate random-walk features to use in the ranking model. This approach is similar to graph-based features used by Agichtein et al. in predicting the quality of contributions in a community question answering site [1] or the email message and author similarity measures developed by Minkov and Cohen [56]. We may also wish to consider behavioral measures previously used to assess the quality of authors contributions to the online community [29].

Proposed Work

A feature-based machine learning approach to leveraging collection structure in ranking presents an interesting avenue of research. We propose three tasks in order to assess the efficacy this approach:

1. **Feature Development** (required): The features presented in Table 7 are only a sample of the possible features we may consider using in a feature-based ranking model. As proposed work, we will expand this set of features potentially useful for ranking, considering several classes of features: query-independent vs. query dependent, features generated from different objects or structural elements of the collection, and random-walk features based on a graph constructed from these elements.

2. **Algorithms Selection & Development** (required): Given the features, we must then select one or more rank-learning algorithms to use in our experiments. Several algorithms are available publicly or internally such as RankSVM [32] and the
Committee Perceptron [27]. Others, such as RankNet [14] would require implementation. As proposed work, we will implement as necessary and apply several rank learning algorithms to this task, investigating both linear ranking models (such as the Committee Perceptron) and non-linear models (such as RankNet).

3. Ablation Studies (required): With a categorized feature set (as in Table 7), we can train and test our rank learning algorithms using subsets of the features in order to investigate each feature category’s effect on ranking performance.

As part of the proposed work, we will perform a series of ablation studies to study the effect of different feature classes on ranking. By evaluating the effectiveness of different feature classes on the overall algorithm performance, we can understand the significance of different aspects of collection structure. This analysis is necessary to validate our original hypotheses.

4.2 Collection Development

Many information retrieval test collections exist, but few contain the rich collection structure we propose to study. As part of this thesis, we will create an information retrieval test collection that can support the investigation proposed here. An information retrieval test collection consists of three components: the document collection, a set of queries, and relevance judgements corresponding to those queries. We describe below our plans for constructing each of these components.

4.2.1 Limitations of Existing Test Collections

Several TREC test collections contain some level of collection structure. For example, we have used the BLOG-06 collection [45] to evaluate the the effectiveness of ranking algorithms that model the relationship of a post to its blog (described in Section 3.1). A portion of the W3C collection [20], used for two years in the TREC Enterprise Track [21, 68], contains archives of public email lists. This collection contains almost 200,000 messages, organized into roughly 130,000 threads. Although these collections offer some level of structure, we are also interested in the wide applicability of the techniques proposed here, particularly to larger collections and those that foster a more rich social interaction. For this reason, we aim to develop a test collection specifically targeted at evaluating collections structure in ranking.

We have begin the development of such a test collection using data from a large and active online forum. This work is described above in Section 3.2.2. Although some preliminary results with this test collection are promising, we believe that the shallow nature of the of relevance information limits the utility of this test collection. For this reason, we propose the following ways to improve the test collection.

4.2.2 Document Collection

In the age of abundant data available online, obtaining documents is the most straightforward piece in building an information retrieval test collection. As described in Section 3, we performed a (nearly) exhaustive crawl of the MacRumors Forum, and were able to

---

16At the time of this writing, the number of contributing authors and mailing lists is unknown
extract the rich structure and metadata present there. Additionally, we have performed a similar crawl of an online forum hosting discussions relating to gardening, the GardenWeb forum\textsuperscript{17}. This forum has fewer messages, roughly the same number of threads, and more users and sub-forums than the MacRumors dataset. See Table 8 for statistics relating to this dataset’s size.

<table>
<thead>
<tr>
<th>Number of Messages</th>
<th>1,854,358</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Threads</td>
<td>252,566</td>
</tr>
<tr>
<td>Number of Sub-forums</td>
<td>224</td>
</tr>
<tr>
<td>Number of Unique Users</td>
<td>98,236</td>
</tr>
<tr>
<td>Post Date Range</td>
<td>February 1996 - October 2008</td>
</tr>
</tbody>
</table>

Table 8: GardenWeb.com Forum Dataset Statistics.

### Proposed Work

This portion of the collection development is complete, and there are no proposed additional tasks.

#### 4.2.3 Queries

Our goal in this task is to identify at least 100 queries for each online forum dataset to use in evaluation. Obtaining (or generating) a realistic set of queries for an information retrieval test collection can be a challenge without direct access to query logs or other user interaction data. One method of identifying information needs for the MacRumors forum was described in section 3, however this method is not without its flaws. The most prominent flaw is that the information needs identified in the MacRumors forum dataset are in the form of natural language messages posted to that online forum. In order to use these in a traditional information retrieval evaluation, we must somehow translate those messages into keyword queries. Several methods have been proposed to extract key phrases \cite{69}, deal with long queries \cite{10}, or generally convert natural language text into search queries \cite{17}. But, this process adds noise to the query generation and may not produce realistic queries that a search system would receive.

For this work, we propose mining real query logs in order to identify queries likely to have relevant documents in the crawled collections. The AOL query log dataset is one such resource in which we may find queries and domain-level click information \cite{61}\textsuperscript{18}. A preliminary assessment of this query log indicates that it contains more than 2000 queries resulting in clicks on a page in the GardenWeb forum, and almost 200 queries resulting in clicks on a page in the MacRumors forum. Although a click-on result does not necessarily indicate that result is relevant, it is an indication that the result was both ranked highly by the AOL search engine as a result of the query and viewed as potentially useful by the user who issued the query. We make the assumption that the queries mined in this way

\textsuperscript{17}http://forums.gardenweb.com/forums/

\textsuperscript{18}This data source was released on August of 2006 and removed shortly after. Although the data is no longer being distributed by AOL, the dataset is widely available through other sources and has been cited frequently in research publications. See, for example: http://portal.acm.org/citation.cfm?id=1146847.1146848#citedby
are a reasonable approximation to queries that may be received in a forum-specific search engine.

Once queries that resulted in clicks in the online forums are identified, we will likely need to manually prioritize and filter those queries in order to ensure that they are reflective of data in our collection. We will apply several criteria for filtering queries, favoring queries that:

- occur more than once in the query log,
- were issued by more than one searcher,
- resulted in more than one click in the domain of interest and
- do not contain clearly offensive or incomprehensible content.

Due to the limited number of potential query candidates in the AOL query log (less than 200 in the case of the MacRumors forum), it is unlikely that all of these criteria will be met while still retaining an adequate number of queries for evaluation. Nonetheless, a best-effort approach will be taken in order to maximize the number of queries to use for evaluation, while eliminating queries less likely to be representative of the task.

**Proposed Work**

(required) This task consists of two components. First, a query set will be identified by mining the AOL query log dataset. Next, this query set will be manually filtered and prioritized to identify the most common queries and most likely to have relevant content in the crawled document collections.

Note that other sources of user queries will be considered should they become available.

### 4.2.4 Document Pooling

Typically in the creation of information retrieval test collections, retrieval results from a variety of systems are pooled together. From this pool of documents, those most likely to be relevant are selected for assessment [71]. For our test set creation, we plan to approximate this pooling process by producing ranked retrieval results from different ranking algorithms and combining those results.

We have already applied six retrieval algorithms to thread search in online forums, as described in Section 3.2. Their performance indicates that these algorithms are moderately diverse. Additionally the mean Kendall’s $\tau$ correlation of the results (averaged across queries and retrieval algorithms) is quite low, $\tau = 0.112(\pm 0.008)$.

**Proposed Work**

(required) For the purposes of identifying documents to be assessed for relevance, we propose to pool the results from those already implemented and tested retrieval algorithms. We will select the top 100 messages threads after pooling per query for assessment.
4.2.5 Relevance Assessment

The final piece in building an information retrieval test collection is the assessment of documents for relevance with regard to the query. For this work we propose using Amazon Mechanical Turk\(^{19}\) (AMT) for relevance assessment. AMT is an online service that provides access to an anonymous pool of *workers*, sometimes referred to as *Turkers*, who select and perform tasks created by *requesters*. These *Human Intelligence Tasks* (HITS) are typically designed to be completed within the confines of a web browser interface. Upon completion of the tasks, the work is assessed by the requester and payment is provided.

Via this huge de-centralized workforce, large volumes of data can be annotated for a small fee, a process often referred to as “crowd-sourcing”. AMT is becoming increasingly popular in research involving many types machine learning and language processing tasks. Recently, several studies have been published using AMT for relevance assessment in information retrieval evaluations \[2, 4, 78\].

Proposed Work

(required) We plan to use AMT for relevance assessment, and will model our process after previous work that has used AMT for this purpose \[4\]. In that study, the authors gathered 4-level relevance assessment (not relevant, marginally relevant, relevant, highly relevant) paying one cent per query/document pair judged.

4.3 Tasks and Test Collections

The discussion of retrieval algorithms focused on the task of thread ranking in online forums, but that is not the only retrieval task that may benefit from leveraging collection structure. In addition to this retrieval task, we are interested in adapting these retrieval models to other domains which may exhibit similar structure. Two other information retrieval test collections produced through previous TREC evaluations also have similar characteristics.

The first of these collections is the W3C collection \[20\] used in the TREC Enterprise track \[21, 68\]. A portion of this collection consists of archives of public mailing lists crawled from the World Wide Web Consortium (W3C) website\(^{20}\), and several message retrieval tasks were run. See Section 2.4.1 for a description of the tasks and an overview of previous approaches. The second collection is the BLOG-06 collection \[45\] used in the TREC Blog track \[47, 60\]. Several tasks were run in the TREC blog track, including blog feed search and blog post search. See Section 3.1 for a description of our approach to the blog feed search task.

Both of these document collections exhibit some level of collection structure. The BLOG-06 collection contains approximately 3 million posts belonging to 100,000 blogs. The W3C collection contains roughly 200,000 messages, with authorship and thread information available through the message metadata. As in online forum search, we hypothesize that search within these collections can also be significantly improved by utilizing this collection structure.

Although these collections contain somewhat different structural elements and the tasks are not identical, we expect that the general approaches to developing retrieval algorithms

\(^{19}\)http://mturk.com

\(^{20}\)http://w3c.org
outlined above are equally applicable to these other tasks. Adapting these techniques will involve identifying which features or elements of the retrieval algorithm can be applied to the task and collection of interest, and whether there are other features unique to the collection that may be useful. For example, in the case of search over the blog collection, we do not have a high-level topical classification of documents as in the online forum datasets. Therefore, features involving the sub-forum do not have direct analogs for retrieval tasks in the blog collection.

Proposed Work

As part of the proposed work, we plan to adapt the above retrieval models to a variety of tasks and collections. A listing of the evaluation collections and tasks that will be investigated over the course of this thesis work is given in Table 9. For each of these tasks, we will do the following:

1. **Adapt Probabilistic Ranking Algorithm** (required): The ranking algorithm presented in Section 4.1.1 leverages several structure elements in online forums. Similar structure may be present in other collections, for example thread and authorship information in the W3C collection. As proposed work, we will modify the algorithm as necessary to other tasks and collections.

2. **Adapt Feature Set** (required): The approaches to feature generation and assessment outlined in Section 4.1.2 are sufficiently general to be applicable to other retrieval tasks. As proposed work, we will modify and extend the feature set as necessary to other tasks and collections.

3. **Experiments** (required): Through ablation studies and parameter tuning, we will assess the value of various feature sets to the tasks under consideration.

### 4.4 Prediction Tasks

In addition to the retrieval tasks described above, exploration of several classification tasks may shed insight into methods of modeling expertise, as well as the general dynamics of these conversational social media collections. In this section we describe some of those tasks.

Although the investigation of these tasks would contribute to the overall understanding of conversational social media, they are not on the *critical path* of the thesis completion.
These tasks should be considered supplementary to the core investigation and will only be addressed as time permits.

Many of these tasks can be thought of as automating some aspects of online forum administration or generally supporting the community interaction. For example, messages could be evaluated at the time they are posted to identify whether the correct sub-forum has been assigned. Then, the results of that evaluation may result in administrative action such as moving the message to the appropriate sub-forum. Given the temporal nature of these online communities, training/test splits should be made along this temporal dimension, with the training messages being posted prior to the test messages.

**Response Prediction**

In conversational social media collections, messages from a variety of authors are typically organized into message threads. These threads contain a start message and zero or more response messages. Given this interaction, several prediction tasks involving the message responses may be of interest to participants in the discussion.

1. **Response Volume Prediction**: Given a thread starting message, can we predict whether that message is likely to produce a large volume of discussion? If popular threads can be identified early, these threads could be promoted on the online forum home page, for example.

2. **Non-Response Prediction**: Given a thread starting message, can we predict whether this message is likely to get zero responses? This is a interesting special case of the previous prediction task. If a user’s message is unlikely to get any responses, a system may inform the user so that action can be taken.

3. **Respondent Prediction**: Given a thread starting message, can we predict which users will respond? If the likely respondents can be identified, those users could be informed by the system of this new thread.

**Sub-forum Classification**

Online forums are generally organized into sub-forums. This sub-forum organization can be thought of as a topical classification of the message threads in the collection. There is often noise in this classification when messages are posted to an unrelated sub-forum. A forum administrator may be interested in a message classification system that can automatically detect when a thread starting message is posted in an incorrect sub-forum.

**Link Prediction**

Within-forum linking is common in some online forums. Given a thread starting message, we may want to predict which threads in the forum are likely to be linked-to by that message. This is similar to the method of identifying questions and relevant threads in Section 3.2.2, however not limited to a question/answer interaction. This task could be used to find “related threads” in the collection, which may be integrated into the forum browsing interface.
5 Tentative Schedule

Table 10 presents a tentative schedule for the completion of the proposed thesis work. Note that the potential areas of investigation are not listed in this schedule. They will performed if time permits.

<table>
<thead>
<tr>
<th>Period</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>present - June 2009</td>
<td>Collection Development</td>
</tr>
<tr>
<td>present - August 2009</td>
<td>Probabilistic Algorithm Development</td>
</tr>
<tr>
<td>July 2009 - October 2009</td>
<td>Feature Engineering and Development,</td>
</tr>
<tr>
<td></td>
<td>Machine Learning Algorithm</td>
</tr>
<tr>
<td>present - January 2010</td>
<td>Retrieval Experiments and Evaluation</td>
</tr>
<tr>
<td>February 2010 - April 2010</td>
<td>Thesis Write-up</td>
</tr>
<tr>
<td>April 2010</td>
<td>Defense</td>
</tr>
</tbody>
</table>

Table 10: Tentative Schedule

Further details on each of the tasks listed in this table are given throughout Section 4.
References


45


46


Craig Macdonald and Iadh Ounis. The TREC blogs06 collection: Creating and analysing a blog test collection. Department of Computer Science, University of Glasgow Tech Report TR-2006-224, 2006. URL http://ir.dcs.gla.ac.uk/terrier/publications/macdonald06creating.pdf. 3.1.4, 4.2.1, 4.3


[68] Ian Soboroff, Arjen P. de Vries, and Nick Craswell. Overview of the TREC-2006 enterprise track. Proceedings of the Fourteenth Text REtrieval Conference (TREC 2006), 2006. 2.3.3, 2.4.1, 4.2.1, 4.3


[70] C J van Rijsbergen. Information Retrieval. Butterworth-Heinemann Newton, MA, USA, 1979. 2.3.1


