Retrieval and Feedback Models for Blog Feed Search

Jonathan L. Elsas, Jaime Arguello, Jamie Callan and Jaime G. Carbonell
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{jelsas, jaime, callan, jgc}@cs.cmu.edu

ABSTRACT

Blog feed search poses different and interesting challenges from traditional ad hoc document retrieval. The units of retrieval, the blogs, are collections of documents, the blog posts. In this work we adapt a state-of-the-art federated search model to the feed retrieval task, showing a significant improvement over algorithms based on the best performing submissions in the TREC 2007 Blog Distillation task[12]. We also show that typical query expansion techniques such as pseudo-relevance feedback using the blog corpus do not provide any significant performance improvement and in many cases dramatically hurt performance. We perform an in-depth analysis of the behavior of pseudo-relevance feedback for this task and develop a novel query expansion technique using the link structure in Wikipedia. This query expansion technique provides significant and consistent performance improvements for this task, yielding a 22% and 14% improvement in MAP over the unexpanded query for our baseline and federated algorithms respectively.

Categories and Subject Descriptors

H.3.3 [**Information Storage And Retrieval**]: Information Search and Retrieval—*Retrieval models,Relevance feedback*

General Terms

Algorithms

Keywords

blog retrieval, federated search, query expansion

1. INTRODUCTION

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.[1]

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.

A second distinguishing characteristic of this task lies in the language of blog posts. Unlike a corpus of newswire stories where each document is a concise and topically coherent unit of text, blog posts are less edited in general and can be more conversational and rambling. Additionally, blog post pages often include reader-generated commentary which has the potential to dramatically inflate the length of the post without necessarily adding any topically related text. Blog collections are particularly susceptible to spam in the form of spam blogs (a.k.a. splogs) and blog comment spam. These are typically machine-generated blogs or blog comments that serve to advertise commercial products and services [9]. These factors combine to make the retrieval task more difficult and may render pseudo-relevance feedback useless. We will show that, without relevance information, standard attempts to extract useful terms or phrases from the blog text uniformly fail.

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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGIR'08, July 20–24, 2008, Singapore.
Copyright 2008 ACM 978-1-60558-164-4/08/07 ...$5.00.

[1] In 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.

previously developed for federated search to blog retrieval, the ReDDE algorithm proposed by Si and Callan [17]. Contrary to previous work in feed retrieval [1, 7, 16], 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.

We also explore the efficacy (or lack thereof) of traditional query expansion techniques such as pseudo-relevance feedback (PRF) for this task. We show that standard PRF on the target corpus fails to improve performance in this task and propose a novel query expansion technique using the Wikipedia[2], a richly linked and edited external corpus.

2. FEED SEARCH RETRIEVAL MODELS

Previous research into feed search models has drawn analogies between this task and several other well-studied retrieval tasks: expert finding [8], cluster-based retrieval [16] and resource selection in distributed information retrieval [7]. All of these tasks share the common goal of ranking collections of documents rather than single documents.

Expert finding, introduced as a TREC task in 2006, is the task of ranking candidate people as potential subject matter experts with respect to a user's query [18]. The experts are typically represented by their collection of email correspondence, and the retrieval models assume that expert candidates would have a large volume of email relevant to the query. In comparison to the blog retrieval task, blog entries are analogous to email messages and blogs to candidate experts. Hannah et. al. [8] applied expert search voting models to blog retrieval and additionally attempted to favor blogs exhibiting query-independent topical "cohesiveness", measured by the average entry distance from the blog's term distribution centroid.

Cluster-based retrieval aims to pre-process the document collection into topical clusters, rank those clusters in response to a query, and then retrieve documents from the highly ranking clusters. One model was recently applied to the problem of blog feed search by Seo & Croft [16]. In their model, each blog feed is a cluster and clusters are ranked by the geometric mean of the query-likelihood of the top-K documents from each cluster.

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. The ReDDE algorithm, a state-of-the-art resource ranking algorithm, was applied to the task of blog retrieval in [1, 7]. Using sampled collection statistics, this algorithm ranks document collections by the estimated number of relevant documents.

Two of the above approaches [1, 7, 16] evaluated against baseline models representing feeds as single "large documents". In these studies, the large document approach consistently outperformed an entry-retrieval approach in which an entry ranking is aggregated into a feed ranking. As we show below, this is not necessarily the case with an appropriate entry-retrieval method that models the topical relatedness of feeds and their entries.

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 federated search. Throughout this paper, we follow the variable naming conventions in Table 1.

Varible	Description
Q, E, F, C	Query, Entry, Feed, Collection
q_i, ψ_i	Query terms, query features
$tf_{t_i;M}$	Frequency of term (or feature) t_i in document M
$\lvert M \rvert$	Size (number of terms or features) of document M
N_C	Number of documents in collection C
N_F	Number of entries in feed F

Table 1: Variable naming conventions.

2.1 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 [7, 1], 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 [3]. 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

$$\begin{aligned} P_{LD}(F|Q) &= P_{LD}(Q,F)/P(Q) \\ &\overset{rank}{=} \underbrace{P(F)}_{\substack{\text{Feed}\\ \text{Prior}}} \quad \underbrace{P_{LD}(Q|F)}_{\substack{\text{Query}\\ \text{Likelihood}}} \end{aligned}$$

where the query likelihood component is estimated with Dirichlet-smoothed maximum likelihood estimates [19]

$$\begin{aligned} P_{LD}(Q|F) &= \prod_{\psi_i \in \Psi(Q)} P_{LD}(\psi_i|F)^{w_i} \\ &= \prod_{\psi_i \in \Psi(Q)} \left(\frac{tf_{\psi;F} + \mu P_{MLE}(\psi|C)}{|F| + \mu} \right)^{w_i} . (1) \end{aligned}$$

The $\psi_i \in \Psi(Q)$ are query features as used in Metzler's full dependence model[13] (query term unigrams and term windows) and μ 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[14, 15]. Our implementation of this retrieval model is described with the following Indri[3] query template

```
#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 [13].

[2] http://en.wikipedia.org

[3] http://www.lemurproject.org/indri/

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 2.2.3 for a detailed explanation of feed priors and how they are used in our models.

2.2 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. 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

$$\begin{aligned} P_{SD}(F|Q) &= \frac{1}{P(Q)} \sum_{E \in F} P_{SD}(Q,E,F) \\ &\overset{rank}{=} P(F) \sum_{E \in F} P(Q|E,F) P(E|F) \\ &\overset{rank}{=} \underbrace{P(F)}_{\substack{\text{Feed}\\\text{Prior}}} \sum_{E \in F} \underbrace{P(Q|E)}_{\substack{\text{Query}\\\text{Likelihood}}} \underbrace{P(E|F)}_{\substack{\text{Entry}\\\text{Centrality}}} \quad (2) \end{aligned}$$

where the last line holds if we assume queries are conditionally independent of feeds given the entry.

The model above extends the one proposed in [1, 7], which are based on the ReDDE federated search algorithm[17]. 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.

2.2.1 *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 [19] rather than Dirichlet (Equation 1), enabling us to combine evidence from the entry, feed and collection

$$\begin{aligned} P_{JM}(Q|E) &= \prod_{\psi_i \in \Psi(Q)} P_{JM}(\psi_i|E)^{w_i} \\ &= \prod_{\psi_i \in \Psi(Q)} \Big(\lambda_E P_{MLE}(\psi_i|E) + \lambda_F P_{MLE}(\psi_i|F) \\ &\qquad + \lambda_C P_{MLE}(\psi_i|C)\Big)^{w_i} \quad (3) \end{aligned}$$

where $\sum \lambda_* = 1, \lambda_* \geq 0$ and $P_{MLE}(\psi_i|M) = \frac{tf_{\psi_i;M}}{|M|}$. Again, the smoothing parameters λ_* 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.

2.2.2 *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 2, would favor longer feeds.

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

$$P(E|F) = \frac{\phi(E,F)}{\sum_{E_i \in F} \phi(E_i,F)}. \quad (4)$$

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

$$\phi_{GM}(E,F) = \prod_{t_i \in E} P(t_i|F)^{P(t_i|E)} = \left(\prod_{t_i \in E} P(t_i|F)^{\frac{tf_{t_i;E}}{|E|}} \right) \quad (5)$$

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 5 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 [8]. 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 4, 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 via a Dirichlet-like smoothing, adding some unobserved centrality mass α

$$P(E|F) = \frac{\phi(E,F) + \alpha\phi(E,C)}{\alpha + \sum_{E_i \in F} \phi(E_i,F)}.$$

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.

2.2.3 *Feed Prior*

The feed prior component, $P(F)$, provides a way to integrate query-independent factors into the feed ranking. Previous uses of document priors in the Indri retrieval framework include favoring documents with shorter URLs in homepage finding tasks, higher PageRank values in web search tasks or higher "signal-to-noise" ratios [15, 2]. 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 equivalent to the ReDDE model if we use the constant entry centrality measure, ϕ_{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.

2.3 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[11, 12], using only the topic title text. 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 (permalinks) 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 `<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 1 and 3), 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?

Model	Prior	Centrality	MAP	P@10
LD	P_{UNIF}	-	0.290	0.400
SD	P_{UNIF}	ϕ_{CONST}	0.277	0.391
SD	P_{UNIF}	ϕ_{GM}	$0.290^{\dagger}$	$0.409^{\dagger}$
LD	P_{LOG}	-	0.188	0.320
SD	P_{LOG}	ϕ_{CONST}	0.298^{+}	0.418^{+}
SD	P_{LOG}	ϕ_{GM}	$\mathbf{0.315}^{\dagger+*}$	**0.424**

Table 2: Mean Average Precision and Precision at 10 for the large document (LD) and small document (SD) retrieval models with different centrality measures (Section 2.2.2) and different feed priors (Section 2.2.3). Statistical significance at the 0.05 level is indicated by † for improvement from ϕ_{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 (ϕ_{GM}), but without the centrality measure (ϕ_{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 (ϕ_{GM} and P_{LOG}) significantly outperforms the best large document model (P_{UNIF}) and using the centrality measure ϕ_{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. FEED SEARCH & QUERY EXPANSION

This section explores which type of query expansion is appropriate for feed search. Several aspects of feed search differentiate it from other retrieval tasks. First, the blogosphere is notoriously filled with spam blogs (splogs) that exist to either sell context-based advertisement or to promote the ranking of affiliated sites[9]. Second, as previously mentioned, we must identify the appropriate representation granularity. The unit of retrieval (the whole feed vs. the feed entry) affects which terms are considered for expansion and the scoring of those terms. It is possible that entire blog feeds are too large or individual blog entries too small to apply standard query expansion techniques to feed retrieval. Third, feed retrieval queries may represent different types of information needs than those driving ad hoc search. Given the nature of feed search, queries may describe more general and multifaceted topics, likely to stimulate discussion over time. If a query corresponds to a high-level description of some topic, there might be a wide vocabulary gap between the query and the more nuanced and faceted discussion in blog posts.

In this investigation, we first applied pseudo-relevance feedback (PRF), a well known and effective method of query expansion, to the task of feed retrieval. Then we developed a novel query expansion technique that scores anchor text linking to Wikipedia articles ranked highly against the base query.

The algorithms are explained in detail below. In all methods, the final, expanded, query, Q_{final}, is explained by the following Indri query template

$$\texttt{\#weight}(\lambda_{fb} Q_{\text{base}} \ (1 - \lambda_{fb}) Q_{\text{exp}}).$$

The base query, Q_{base}, which retrieves the feedback documents, is the dependence model query constructed from the topic title as described above in Section 2.1. Q_{exp} is a weighted query of the form

$$\texttt{\#weight}(\lambda_1 \texttt{\#combine}(w_1) \ \lambda_2 \texttt{\#combine}(w_2) \ ... \\ \lambda_T \texttt{\#combine}(w_T)),$$

where w_i are expansion terms or phrases and λ_i are the weights assigned by the query expansion algorithm. In all tests, the feedback mixing weight, λ_{fb} is fixed at 0.5. In the final feed ranking, the expanded query Q_{final} is used for the calculation of query likelihood in our retrieval models presented above.

3.1 Pseudo-Relevance Feedback

PRF assumes the top N retrieved documents, D_N, are relevant to the base query and extracts highly discriminative terms or phrases from those document as query expansion terms. The state-of-the-art PRF method used in this work was Indri's built-in PRF facility, an adaptation of Lavrenko's relevance model. The reader is referred to [10] for details of this method.

PRF under several different conditions was applied to the task of feed retrieval to address the following two questions:

- $Q1$: Are either of the two units of retrieval, entire blogs and single blog posts, suitable for query expansion? Is one better than the other?

- $Q2$: Is the unedited language in blogs concise and topical enough for PRF to work, or is it too diluted?

$Q1$ is a matter of representation (i.e., the unit of analysis adopted during PRF) and $Q2$ is a matter of content (i.e., the language in blogs).

The following PRF variants were evaluated.

- `PRF.FEED`: PRF where D_N are the top N feeds.

- `PRF.ENTRY`: PRF where D_N are the top N feed entries.

- `PRF.WIKI`: PRF where D_N are the top N Wikipedia articles when the base query is run on the Wikipedia.

- `PRF.WIKI.P`: PRF where D_N are the top N Wikipedia passages (sequences of at most 220 words, the average entry length in the BLOG06 corpus) when the query is run on the Wikipedia.

- `NO_EXP` (baseline): No query expansion.

Note that `PRF.WIKI` is identical to the *external expansion* method developed in [6], where the relevance model is estimated entirely from our external corpus. In addition, we evaluated two (true) relevance feedback (RF) methods.

- `RF.TTR`: RF where expansion terms originate from the top 10 relevant feeds (`TTR` = "top 10 relevant"). `RF.TTR` simulates the senario where a user judges documents down the ranking until 10 relevant documents are found. Expansion is always done.

- `RF.RTT`: RF where expansion terms originate from the relevant feeds within the top 10 (`RTT` = "relevant in top 10"). This method simulates the scenario where a user inspects the top 10 documents and feedback terms are pulled from those found to be relevant. Expansion is done only if at least one of the top 10 is relevant.

For all PRF-based methods, $N = 10$ and $T = 50$. These values were previously found to produce positive results[14].

3.1.1 PRF Results for Blog Feed Retrieval

Complete PRF results are shown in the upper-half of Table 3. To address $Q1$, `PRF.ENTRY` performed slightly better than `PRF.FEED`. However, neither representation improved upon the unexpanded query, `NO_EXP`. A natural question is whether the typical blog entry is too small or the typical blog feed too large for PRF to work. The average blog entry in our BLOG06 index has 220 words. To determine if the typical size of an entry is too short for PRF, `PRF.WIKI.P` was set to perform PRF on the Wikipedia using as documents (non-intersecting) passages of 220 words of length. `PRF.WIKI.P` significantly outperformed `NO_EXP` in terms of MAP and P@10 for the **LD** model, showing that it is not the size of entries that makes `PRF.ENTRY` ineffective. Furthermore, both `RF.RTT` and `RF.TTR` (gray, italicized in Table 3) significantly outperformed `NO_EXP`, showing that, if the relevant feeds are known, they are not too large for PRF to work. Therefore, both the entry and the feed have the potential to be effective for PRF.

To address $Q2$, the improvement in performance of `RF.RTT` and `RF.TTR` over `NO_EXP` is also informative. If the relevant feeds are known, it is not the case that the posts are too noisy or the discussion too vague or diluted to allow effective expansion terms to be chosen by a method such as PRF (using feeds). Thus, PRF could potentially yield a significant improvement over no expansion.

The above results also show the Wikipedia's potential as a valuable source of expansion terms for feed search queries. In the **LD** model, `PRF.WIKI` and `PRF.WIKI.P` both outperformed `PRF.ENTRY` and `PRF.FEED`, yielding significant improvement over `NO_EXP`. Spam was an issue for both `PRF.FEED` and `PRF.ENTRY`, but more of a problem for `PRF.ENTRY`. For example, `PRF.ENTRY` added pornography-related expansion terms to 8/45 queries, whereas `PRF.FEED` added pornography-related terms to only 3/45 queries.

Surprisingly, the **SD** model is not improved upon by these Wikipedia query expansion techniques. The next section introduces a novel hyperlink-based query expansion technique, effective across retrieval models, which runs the base query against the Wikipedia and scores anchor phrases found in hyperlinks pointing to highly ranked Wikipedia articles.

3.2 Wikipedia Link-based Query Expansion

Our Wikipedia index consists of $2,471,311$ documents, excluding date and category pages, from the English Wikipedia. The original Wikipedia markup includes useful metadata such as cross-article hyperlinks. Each hyperlink is specified by the title of the target Wikipedia article and optional anchor text. When specified, the anchor text provides an alternative description of the target article's title (e.g., "US" → "United States of America").

Our simple link-based query expansion technique begins by running the base query on the Wikipedia. From the resulting ranking of Wikipedia articles, two sets are defined. The relevant and working sets, S_R and S_W, contain articles ranking within the top R or top W retrieved results. Constraining $R < W$ implies that $S_R \subset S_W$.

Then, each anchor phrase, a_i, occuring in an article in S_W and linking to an article in S_R is scored according to

$$\lambda_i = score(a_i) = \sum_{a_{i_j} \in S_W} \Big(\mathbb{I}(\text{target}(a_{i_j}) \in S_R) \times (R - \text{rank}(\text{target}(a_{i_j}))) \Big),$$

where a_{i_j} denotes an actual occurrence of anchor phrase a_i, function target(a_{i_j}) returns the target article of the hyperlink anchored by a_{i_j}, and $\mathbb{I}(\bullet)$ is the identity function and equals 1 if condition $\bullet$ holds true. Based on this scoring function, the highest scoring candidate expansion phrases are those that anchor many hyperlinks pointing to articles ranked high against the base query.

The WIKI.LINK expansion method is stable in both the terms selected for query expansion and the retrieval performance across a wide range of R values from 50 to 200. In this work, parameters R and W were set to $R = 100$ and $W = 1000$. T, the number of expansion phrases, was set to $T = 20$. Unlike the PRF-based methods, PRF.*, this algorithm finds expansion (multi-term) phrases rather than single terms. 20 phrases were extracted so that the total number of expansion terms would be roughly equivalent to the 50 expansion terms allowed for the PRF-based methods. Note that this algorithm assumes nothing about the underlying retrieval model.

Because WIKI.LINK focuses only anchor phrases, this query expansion technique considers many fewer, but potentially higher quality, expansion terms and phrases than other query expansion methods. In our experiments with $R = 100$, on average WIKI.LINK only considered approximately 200 phrases for query expansion per query, whereas using the top 10 documents from Wikipedia in PRF.WIKI considered approximately 9000 terms.

With this novel algorithm, we address the following questions.

- $Q3$: Does WIKI.LINK outperform PRF for feed search? Why or why not?
- $Q4$: Does WIKI.LINK generalize across retrieval tasks, collections or models? Why or why not?

3.2.1 Linked-Based Query Expansion Results

To address questions $Q3$ and $Q4$, we applied WIKI.LINK to TREC queries 951-995 from the TREC 2007 Blog Distillation Task using the best large document and small document models (BD07.LD and BL07.SD), as well as to ad hoc TREC queries 701-750 from the TREC 2004 Terabyte Track (TB04) and queries 751-800 from the TREC 2005 Terabyte Track (TB05)[4, 5]. For comparison, the standard PRF methods from Section 3.1 were also applied the to ad hoc search query sets TB04 and TB05. Again, $N = 10$ and $T = 50$ for all PRF-based methods. Results are shown in Table 3 in terms of these three query sets.

Blog Feed Retrieval

	MAP		**P@10**	
	BD07.LD	BD07.SD	BD07.LD	BD07.SD
NO_EXP	0.289	0.315	0.393	0.424
PRF.FEED	0.272	0.266	0.389	0.393
PRF.ENTRY	0.290	0.282	0.384	0.391
PRF.WIKI	0.327*	0.314	0.458*	0.431
PRF.WIKI.P	0.319*	0.313	0.433*	0.416
WIKI.LINK	**0.355***†	**0.361***†	**0.476***†	**0.482***†
RF.RTT	*0.386*	*n/a*	*0.536*	*n/a*
RF.TTR	*0.366*	*n/a*	*0.439*	*n/a*

Ad hoc Retrieval

	MAP		**P@10**	
	TB04	TB05	TB04	TB05
NO_EXP	0.294	0.354	0.543	0.608
PRF	**0.317**	**0.380**	**0.582**	0.602
PRF.WIKI	0.311	0.345	0.578	**0.632**
PRF.WIKI.P	0.306	0.352	0.570	0.616
WIKI.LINK	0.258	0.320	0.519	0.580

Table 3: MAP and P@10 for multiple expansion & retrieval models, and different test sets. On the BD07 tests, significance at the 0.05 level over the NO_EXP and PRF.WIKI models is indicated by $*$ and †, respectively.

In response to question $Q3$, our link-based approach, WIKI.LINK, outperformed all PRF-based methods, including PRF.WIKI in the Blog Distillation task. Additionally, WIKI.LINK seems to be more robust, helping more and hurting fewer queries than PRF.WIKI. For example using the **LD** model, WIKI.LINK improved average precision by at least 25% on 15/45 queries and only hurt performance on 8/45 queries. In contrast, PRF.WIKI improved performance on 11/45 queries by at least 25% and hurt performance on 11/45 queries. These findings generalize across the retrieval models presented above.

In response to question $Q4$, it's worth noting that none of the Wikipedia-based methods perform as well on query sets TB04 and TB05 as they do on query set BD07. This could be partially attributed to the target corpus. TREC ad hoc queries 701-800 (TB04 & TB05) are run on the GOV2 corpus, composed of documents pulled from the .gov domain where spam is not an issue. A cleaner external corpus, such as the Wikipedia, has potentially less to contribute as a source of expansion terms for these query sets because the target corpus is already relatively clean.

What remains to be answered is the "Why?" part of questions $Q3$ and $Q4$. Why does WIKI.LINK work on the blog search task but not generalize to the ad hoc retrieval task? Compared to PRF.WIKI, WIKI.LINK may be more heavily biased towards finding expansion terms that cover a broader topical range for two reasons. First, it considers candidate expansion phrases related to the top 100 ranked articles, while PRF.WIKI considers terms from the top 10. Second, it focuses on anchor phrases, each describing the title of a Wikipedia article. Because the Wikipedia is an encyclopedic resource and each Wikipedia page has a distinct topical focus, considering anchor text biases the algorithm towards expansion phrases covering a wide topical range. Thus, it's possible that a general query stands to gain more from our WIKI.LINK expansion method. A very specific query could be negatively impacted by WIKI.LINK's bias towards wide-coverage expansion terms.

Our hypothesis is that feed search queries tend to describe more high-level, multifaceted topics than those described by ad hoc queries. These more general topics stand to gain more from expansion using an external encyclopedic resource such as the Wikipedia. General topics tend to have many relevant articles in the Wikipedia, each relevant because it covers a unique facet of the high-level topic. Under these conditions, hyperlinks cross-referencing these many relevant subtopic documents become an effective resource.

3.3 Investigating Query Generality

To test our hypothesis that feed search queries are, on average, more general than ad-hoc queries, we formulated 3 simple tests of generality. Each test assumes that a simple measure can be used as a proxy for generality/specificity. Although in isolation each test measures only one aspect of generality, taken together, they reach the same conclusion: feed retrieval queries are more general than ad hoc retrieval queries. Again, we applied the three tests to our three query sets used above (TB04, TB05 and BD07).

3.3.1 Test 1: Query Length

This test makes the assumption that a longer query is more specific than a shorter one. For example, TREC query 710, "prostate cancer treatments" is more specific than TREC query 959, "bipolar disorder", since the first one focuses on a specific aspect (the "treatment") of the health condition, while the second focuses on all aspects of the health condition. Of course, exceptions are not hard to find, TREC query 715, "schizophrenia drugs" is not more general than "prostate cancer treatments". The assumption, however, is that usually more words mean more modifiers that focus query on a more specific topic. Table 4 shows the average query length of the queries in each of our three sets.

TB04	**TB05**	**BD07**
3.16	3.08	1.88

Table 4: Test 1. Average Query Length.

The difference between BD07 and both Terabyte Track query sets (TB0*) is statistically significant according to an unpaired, 2-tailed, t-test ($P < 0.005$ in both cases). The difference between the TB04 and TB05 is not significant.

3.3.2 Test 2: ODP Depth

The Open Directory Project[4] (ODP) is a directory of the web maintained by volunteer editors. The directory takes the form of a topic hierarchy with general topics (e.g.,"Sports") closer to the top of the hierarchy and more specific topics further down the tree (e.g., "Sports" → "Olympics" → "Beijing 2008"). Each ODP node webpage is composed of links to neighboring ODP nodes (mostly children nodes) and links to pages assigned to it. Each reference to a webpage assigned to a node shows the document's title and a short summary. Thus, each ODP node contains text, in the form of summary snippets, that describe the documents assigned to it.

We make the assumption that, when run against the ODP, a general query will tend to return documents corresponding to nodes higher up in the ODP tree (more general nodes). Using the Yahoo! Web API[5], each query was run constrained to only return documents within the ODP domain. Then, the average document ODP depth among the top 10 documents was computed. Finally, the query average ODP depth was averaged across all queries in the same set (Table 5). On average, an ODP node returned in response to a Blog Feed 2007 query is higher up in the ODP tree.

TB04	**TB05**	**BD07**
5.19	5.29	4.74

Table 5: Test 2. Average ODP depth of documents returned by query.

The difference between BD07 and both Terabyte Track query sets (TB0*) is statistically significant according to an unpaired, 2-tailed, t-test ($P < 0.05$ in both cases). The difference between the TB04 and TB05 is not significant.

3.3.3 Test 3: Wikipedia Relevant Set Cohesiveness

Let S_R be the relevant set as defined above. Then, let L denote the set of all hyperlinks appearing in the documents in S_R, $L = \{l | l \in S_R\}$, and L_{in} denote the set of hyperlinks in S_R referencing a document also in S_R, $L_{in} = \{l | l \in S_R, \text{target}(l) \in S_R\}$. The ratio $\frac{|L_{in}|}{|L|}$ can considered a measure of cohesiveness among the documents in S_R. It is the fraction of all hyperlinks in S_R that link back to a document in S_R.

We make the assumption that a general query (e.g, TREC query 960, "garden") will have more relevant documents in the Wikipedia than a specific query (e.g, TREC query 787, "sunflower cultivation"). We also make the assumption that those Wikipedia articles covering the relevant subtopics of the query topic are cross-referenced with hyperlinks. Given these assumptions, the ratio $\frac{|L_{in}|}{|L|}$ should be higher for general queries. Figure 1 shows the average cohesiveness ratio, $\frac{|L_{in}|}{|L|}$, for each query set, varying R from the top 5 to top 100 documents. On average, the top R Wikipedia articles returned in response to a blog feed query are more interconnected than those returned in response to an ad-hoc query and this difference increases with R. This test supports our hypothesis that feed search queries are more

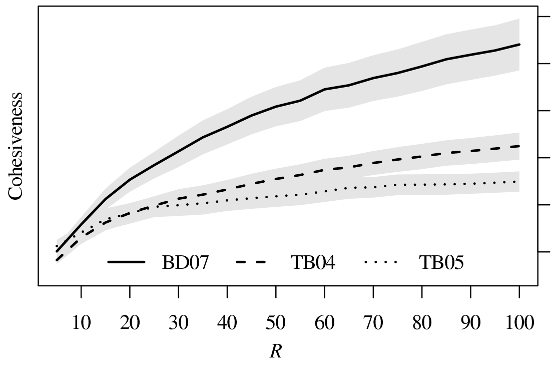

Figure 1: Test 3. Average query set cohesiveness, $\frac{|L_{in}|}{|L|}$, for $R \in [5, 100]$. Shaded region shows one standard error.

[4] http://www.dmoz.org

[5] http://developer.yahoo.com/

general. Furthermore, it shows that feed retrieval queries are especially well suited for WIKI.LINK, as the algorithm can leverage greater evidence, in the form of more in-links, to better estimate candidate expansion term weights.

The three tests above show that the queries used in the Blog Distillation task (BD07) are measurably different than the queries from both Terabyte Tracks (TB04 and TB05). These query differences, however, do not directly predict whether or not the WIKI.LINK query expansion method will improve retrieval performance. None of the three measures exhibit a strong correlation with performance improvement when using this expansion method. This analysis focuses on the query sets from these tasks, but the performance of any query expansion method is a function not only of the query, but also the corpora used, the retrieval algorithms, and term scoring algorithms. Although query generality likely plays a role in the performance of the WIKI.LINK query expansion method, it is not the only factor in predicting expansion performance.

4. CONCLUSION & FUTURE WORK

This work explored the task of blog feed retrieval from two perspectives: retrieval models and query expansion algorthms. 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 [1, 7, 16] 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.

Two aspects of blog feed search that were left unexplored in this work are analysis of linking patterns across the blog corpus and the influence of post timestamping on retrieval. Link-network analysis in the blogosphere is a well studied area and has potential for further improving retrieval performance. Also, current blog post retrieval services strongly favor more recent posts in the ranking algorithms, and temporally profiling a feed's set of entries may lead to further improvements.

In addition to retrieval models, we presented an in-depth analysis of query expansion for blog feed retrieval. On this task, our novel Wikipedia link-based approach obtained a greater than 13% improvement over no expansion (across large and small document models) in terms of both MAP and P@10. Although this method did not generalize to the Terabyte Track ad hoc queries it does show promise for queries that represent more general information needs, similar to those typical of feed retrieval.

5. ACKNOWLEDGEMENTS

This work was supported by NSF grants IIS-0240334 and CNS-0454018 as well as by the Defense Advanced Research Projects Agency (DARPA) under Contract Number IBM W0550432 (DARPA PRIME Agreement # HR0011-06 -2-0001). Any opinions, findings, conclusions, and recommendations expressed in this paper are the authors' and do not necessarily reflect those of the sponsors.

6. REFERENCES

[1] J. Arguello, J. L. Elsas, J. Callan, and J. G. Carbonell. Document representation and query expansion models for blog recommendation. In *Proc. of the 2nd Intl. Conf. on Weblogs and Social Media (ICWSM)*, 2008.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[3] J. Callan. Distributed information retrieval. In W. Croft, editor, *Advances in Information Retrieval*, pages 127–150. Kluwer Academic Publishers, 2000.

[4] C. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2004 terabyte track. In *Proc. of the 2004 Text Retrieval Conf.*, 2004.

[5] C. Clarke, F. Scholer, and I. Soboroff. Overview of the TREC 2005 terabyte track. In *Proc. of the 2005 Text Retrieval Conf.*, 2005.

[6] F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *Proc. of the 29th Annl. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 154–161, 2006.

[7] J. Elsas, J. Arguello, J. Callan, and J. Carbonell. Retrieval and feedback models for blog distillation. In *Proc. of the 2007 Text Retrieval Conf.*, 2007.

[8] D. Hannah, C. Macdonald, J. Peng, B. He, and I. Ounis. University of Glasgow at TREC 2007: Experiments with blog and enterprise tracks with terrier. In *Proc. of the 2007 Text Retrieval Conf.*, 2007.

[9] P. Kolari, A. Java, and T. Finin. Characterizing the splogosphere. In *Proc. of the 3rd Annl. Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 15th World Wide Web Conf.*, 2006.

[10] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proc. of the 24th Annl. Intl. ACM SIGIR Conf. on Research and Development in Information retrieval*, pages 120–127, 2001.

[11] C. Macdonal and I. Ounis. The TREC blog06 collection: Creating and analysing a blog test collection. Technical Report TR-2006-224, Department of Computing Science, U. of Glasgow, 2006.

[12] C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC 2007 blog track. In *Proc. of the 2007 Text Retrieval Conf.*, 2007.

[13] D. Metzler and B. W. Croft. A markov random field model for term dependencies. In *Proc. of the 28th Annl. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 472–479, 2005.

[14] D. Metzler, T. Strohman, H. Turtle, and W. Croft. Indri at TREC 2004: Terabyte track. In *Proc. of the 2004 Text Retrieval Conf.*, 2004.

[15] D. Metzler, T. Strohman, Y. Zhou, and W. Croft. Indri at TREC 2005: Terabyte track. In *Proc. of the 2005 Text Retrieval Conf.*, 2005.

[16] J. Seo and W. B. Croft. Umass at trec 2007 blog distillation task. In *Proc. of the 2007 Text Retrieval Conf.*, 2007.

[17] L. Si and J. Callan. Relevant document distribution estimation method for resource selection. In *Proc. of the 26th Annl. Intl. ACM SIGIR Conf. on Research and Development in Informaion Retrieval*, 2003.

[18] I. Soboroff, A. de Vries, and N. Craswell. Overview of the trec 2006 enterprise track. In *Proc. of the 2006 Text Retrieval Conf.*, 2006.

[19] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214, 2004.