Document Representation and Query Expansion Models for Blog Recommendation

Jaime Arguello and **Jonathan L. Elsas** and **Jamie Callan** and **Jaime G. Carbonell**
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Abstract

We explore several different document representation models and two query expansion models for the task of recommending blogs to a user in response to a query. Blog relevance ranking differs from traditional document ranking in ad-hoc information retrieval in several ways: (1) the unit of output (the blog) is composed of a collection of documents (the blog posts) rather than a single document, (2) the query represents an ongoing – and typically multifaceted – interest in the topic rather than a passing ad-hoc information need and (3) due to the propensity of spam, splogs, and tangential comments, the blogosphere is particularly challenging to use as a source for high-quality query expansion terms. We address these differences at the document representation level, by comparing retrieval models that view either the blog or its constituent posts as the atomic units of retrieval, and at the query expansion level, by making novel use of the links and anchor text in Wikipedia[1] to expand a user's initial query. We develop two complementary models of blog retrieval that perform at comparable levels of precision and recall. We also show consistent and significant improvement across all models using our Wikipedia expansion strategy.

Introduction

Blog retrieval is the task of finding blogs with a principle, recurring interest in X, where X is some information need expressed as a query. The input to the system is a short (i.e., 1-5 word) query and the output is a ranked list of blogs a person might want to subscribe to and read on a regular basis. This was the formulation of the TREC 2007 Blog Distillation task (Macdonald, Ounis, & Soboroff 2007). Feed recommendation systems may also suggest relevant feeds based on the feeds a user already subscribes to (Java *et al.* 2007)[2]. However, in this work, a short query is assumed to be the only evidence of a user's interest. The output is a ranked list of feeds expected to satisfy the information need in a persistent manner and not just with a few relevant entries. We interchangeably refer to this query-in/blogs-out approach to blog/feed recommendation as blog/feed retrieval.

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

[2] In this work, we will refer to "blogs" and "feeds" as the same entity, as there is a one-to-one relationship between the two. "Entry" and "post" will also be used interchangeably.

Blog retrieval differs from traditional ad-hoc retrieval in several important ways. First, the ultimate unit of output (the blog) corresponds to a collection of documents (its blog posts) rather than a single document. A single relevant post does not imply the relevance of its corresponding blog. Therefore, we must be concerned with how relevance at the post level corresponds to relevance at the overall blog level. Second, the nature of relevance at the blog-scale has implications on the expected information needs of the users of a blog retrieval system. If blog authors are expected to have an ongoing interest in a topic, that topic is likely multi-faceted and supports the authors' desire to write posts on various aspects of the central topic. Thus, users' information needs appropriate for a blog retrieval system are likewise multi-faceted. A short query is an impoverished representation of a user's interest in feed recommendation as it does not convey these facets. Finally, a blog corpus is not a typical document collection, but susceptible to large amounts of reader-generated commentary of varying quality and topicality, and large amounts of comment-spam and spam blogs (splogs) intended only to route traffic to desired commercial sources. Any technique used in blog retrieval must be robust to this "noise" in the collection.

Two dimensions of feed retrieval were investigated to address these unique aspects of blog search.

1. **Representation**: How do we effectively represent blogs for use in a retrieval system? In this work, we considered two models of representation: the *large document* model in which entire blogs are indexed as single documents and the *small document* model where we index at the post-level and aggregate a post ranking into a final blog-ranking.

2. **Query Expansion**: Does the nature of this task and the noise in the collection require different techniques for query expansion than traditional ad-hoc retrieval? In typical retrieval systems, query expansion is often intended to overcome a vocabulary mismatch between the query and the document collection. In this task, however, we may be able to view query expansion as bridging the gap between a high-level general topic (expressed by the query) and the more nuanced facets of that topic likely to be written about in the blog posts.

In this work, we develop several representations and retrieval models for blog retrieval and present a novel technique for mining the links and anchor text in Wikipedia for query expansion terms and phrases. The remainder of the paper is organized as follows. First we discuss our models of feed retrieval and query expansion. Our test collection and evaluation setup are discussed next, followed by our experimental results and a brief error analysis. We conclude with a discussion of related work and future directions for this research.

Feed Representation and Retrieval Models

As stated above, the issue of how to represent feeds for retrieval is critical to the task of effectively ranking in response to a query. In this work we explored two primary models of representation for feed retrieval. The first, the "large document model", represents each feed as a single document, a virtual concatenation of its respective entries. The second, the "small document model", represents each entry as an individual document and an entry ranking is aggregated into a feed ranking post-retrieval.

Large document model

The ultimate unit of retrieval is the feed, and for this reason one clear approach is to index feeds as single documents. In this scenario, all posts or entries in the feed are concatenated together to form one large bag of words or phrases. This *large document* approach is appealing for its simplicity: existing retrieval techniques can be easily applied to this feed retrieval without modification. This is similar to the "global representation" approach taken by (Seo & Croft 2007) and a simplified version of the large document model in (Elsas *et al.* 2007). In our experiments we used Indri's[3] language modeling approach to retrieval, ranking documents using the full-dependence retrieval model proposed by (Metzler & Croft 2004; 2005). This formal Markov random field retrieval model fits with Indri's retrieval framework and takes into account dependencies between query terms through ordered- and unordered-window constraints. This extended query is used to estimate the query generation likelihood given a feed, which gives our final relevance scoring function,

$$Rel(F) = P(Q|F). \quad (1)$$

An example full-dependence query for the query string "DSLR camera review" is as follows:

```
#weight( 0.8 #combine( DSLR camera review )
         0.1 #combine( #1( DSLR camera )
             #1( camera review )
             #1( DSLR camera review ))
         0.1 #combine( #uw8( DSLR camera )
             #uw8( camera review )
             #uw8( DSLR review )
             #uw12( DSLR camera review )))
```

where the first line is a unigram query, the second group is a query of ordered windows or exact phrases, and the third group is a query of unordered windows. The parameters used in our dependence model queries (0.8, 0.1, 0.1 and window sizes) are taken directly from (Metzler & Croft 2005), and have been shown to perform effectively across a variety of corpora and tasks. The reader is referred to the above references for a detailed explanation of Indri's language modeling and Markov random field retrieval models.

This large document approach is a straightforward application of existing retrieval techniques to feed retrieval, but it may have some potential pitfalls. First, if some entries in a feed are disproportionately larger than others, the larger posts will dominate that feed's language model. Although we are interested in ultimately ranking feeds in this task, the unit of consumption by an end user is a single post-at-a-time. For this reason, it is critical to not let a single large post bias our relevance scoring of its corresponding feed.

Secondly, when all posts in the feeds are concatenated together, the resulting document collection may have an extremely skewed size distribution. Some feeds may be updated daily or several times a day resulting in large feed documents, whereas others may be updated weekly or monthly resulting in much smaller feed documents. This extreme variance in the sizes of our large documents may make it difficult for existing retrieval algorithms to adequately normalize for document length. These two issues are addressed below with the small document model.

Small document model

In order to accommodate the potential deficiencies of the large document model outlined above, we can treat the entry as the fundamental unit of retrieval and aggregate the entry ranking into a feed ranking, taking care to normalize for the number of entries per feed. In this *small document* model, we can draw an analogy between feed retrieval and the task of resource ranking in distributed information retrieval.

Distributed IR, or federated search, is the problem of searching across multiple, possibly many, different text databases (Callan 2000). It is often formulated as three separate, but interrelated subtasks: (1) collecting information about each collection's contents (*resource representation*), (2) ranking the resources and selecting the few most likely to contain many documents relevant to the query (*resource ranking*), and (3) merging the results from the selected databases into a single document ranking (*results merging*). In the second of these tasks, resource ranking, the goal is to rank higher the databases more likely to contain many documents relevant to the query. Similarly, in feed search the goal is to rank higher the feeds more likely to contain a majority of posts relevant to the query.

Our basic small document approach is closely related to the ReDDE resource ranking formula for federated search proposed by (Si & Callan 2003). In that model, external databases or resources are ranked by their expected number of relevant documents using sampled database statistics. We take a similar approach here, and this can be viewed as a straightforward extension of the large document query likelihood model. In this model, we decomposed the feed

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

relevance scoring into a weighted sum of entry likelihoods:

$$Rel(F) = \sum_{E \in F} \underbrace{P(Q|E)}_{\substack{\text{Query} \\ \text{Likelihood}}} \times \underbrace{P(E|F)}_{\substack{\text{Entry} \\ \text{normalization}}} \quad (2)$$

In the above formulation, the term on the right, entry normalization, serves a dual purpose. The first purpose is to add a means to normalize across feeds with different number of entries. The query likelihood, $P(Q|E)$, should be more influential when E belongs to a feed with few posts than when E belongs to a feed with many posts because E is more representative of the feeds overall content. The second purpose is to add a measure of centrality of the entry to the feed as a whole, favoring entries that have a language more like the rest of the feed. We only considered the first application, modeling $P(E|F)$ as uniform on a per-feed basis, and leave for future work investigating varying this probability on a per-entry basis.

Entry normalization In this work we considered two different methods for normalizing feed lengths. The first is to use the simple uniform maximum likelihood estimate (MLE) of $P(E|F)$:

$$\hat{P}_{MLE}(E|F) = \frac{1}{|F|} \quad (3)$$

where $|F|$ is the number of entries in this feed. This provides a sensible method for normalizing for feed length. But, by using the MLE, we assume we have an accurate sample of all the feeds in our collection. In reality, the creation of the feed corpus is unavoidably constrained because of the time-period of the collection and we therefore have an imperfect view of the true probability $P(E|F)$. For this reason, we consider a second estimation method, smoothing the MLE as follows:

$$\hat{P}_{SM}(E|F) = \frac{1 + \frac{\mu}{\#Entry \in C}}{|F| + \mu} \quad (4)$$

where $\#Entry \in C$ is the number of entries in the collection, about 3.1 million, and μ is a smoothing parameter we fix at the average number of entries per feed for these experiments, $\hat{\mu} \approx 34.3$. This model assumes there are some number, μ, of unobserved (and non-relevant) entries in each feed and therefore penalizes feeds with a small number of observed entries. Intuitively, this may have a potentially beneficial effect: if we only observe a few entries in a given feed, we shouldn't have the confidence to recommend this feed to a user posing a query to our system.

Query likelihood estimation In addition to our two methods of estimating $P(E|F)$, equations 3 and 4, we also considered two methods for estimating the entry query likelihood, $P(Q|E)$. As above in our large document model, we can estimate this using the full dependence retrieval model, running the query with window constraints on entries rather than feeds. This is expected to give reasonable performance, but ignores a potentially critical aspect of feed retrieval: that entries are not independent entities, but rather parts of larger feeds.

In this sense, we can consider entry retrieval as similar to passage retrieval or XML element retrieval. In previous work in these areas (Ogilvie & Callan 2004) it has been shown that retrieval performance can be greatly improved by using hierarchical language modeling of the "children" nodes, in this case entries. This is done by interpolating the passage or element language model with the language model of the larger document. We took a similar approach with entry retrieval, smoothing the entry language model with both the feed ("parent") and collection ("global") language models. For a simple bag-of-words query, we have the following estimate:

$$\hat{P}_H(Q|E) = \prod_{t \in Q} \Big(\lambda_E \hat{P}_{MLE}(t|E) + \lambda_F \hat{P}_{MLE}(t|F) + \lambda_C \hat{P}_{MLE}(t|C) \Big) \quad (5)$$

where the probabilities are straightforward maximum likelihood estimates, for example $\hat{P}_{MLE}(t|E) = \frac{tf_{t;E}}{|E|}$, where $tf_{t;E}$ is the term frequency of term t in the entry E and the λ's are mixing parameters, $\lambda_E + \lambda_F + \lambda_C = 1$. This bag-of-words retrieval model generalizes in a straightforward way to the dependence model described above. We fixed these parameter values at $\lambda_E = 0.5, \lambda_F = 0.2$ and $\lambda_C = 0.3$. These parameters settings are in the range of settings that have worked well in the past for XML element retrieval, although further exploration of refining these parameters estimates is a focus of ongoing research.

Feed Search and Query Expansion

Automatic query expansion (AQE) is a widely used technique in information retrieval. The general sequence of steps is: the system runs the input query, assumes that some fraction of the top-ranked documents are relevant, extracts terms or phrases characteristic of those top-ranked documents (i.e. more likely to occur in the top-ranked documents than in the collection as a whole), adds these terms (possibly weighted somehow) to the original query, and runs the expanded query. Query expansion is often used as a technique to overcome a possible vocabulary mismatch between the query and a relevant document and to broaden the scope of the query while hopefully staying on topic (Manning, Raghavan, & Schütze 2008). Typically, the base query is run on the target collection (i.e., the collection being queried) and expansion terms are pulled from the top N documents. To avoid expanding the query with unrelated terms, N is usually kept small. $N = 10$ has been shown to work well in web retrieval (Metzler *et al.* 2006).

One question explored in this work is whether a query expansion technique designed with feed search in mind performs better than a typical AQE technique employed in ad-hoc search. Feed retrieval may require a different query expansion technique than ad-hoc retrieval on the web for two reasons. The first reason stems from the large volume and unique nature of spam blogs (i.e., splogs) in the blogosphere. The second reason stems from the types of information needs that seem to be typical in feed retrieval.

Spam blogs, or splogs, exist primarily for hosting profitable context-based advertisements or link farms aiming to

increase the rank of affiliated sites (Kolari, Java, & Finin 2006). As opposed to much more static general web spam, a splog must provide new content continuously. This new content is either machine-generated or scrapped from legitimate blogs or web pages. Machine-generated content is easier to detect than text scrapped from genuine resources. (Lin *et al.* 2006) show that incorporating temporal self-similarity measures with respect to the prime attributes of blogs (i.e., content, outgoing links and tags) improves splog detection over just considering content features. Spam blogs can have a negative effect on query expansion by artificially inflating the relative importance of meaningless terms. In prior work, filtering spam blogs improved blog post retrieval performance, especially in the case of commercially-oriented queries (e.g., "Apple iPod") (Mishne 2007).

Second, feed search queries may have a different aim than ad-hoc search queries. (Mishne & de Rijke 2006) examined a set of 1,400 queries submitted to a large blog search engine, Blogdigger.com [4], that supports both *ad-hoc* and *information filtering* queries. In information filtering, the user submits a query, which remains fixed, and gets continuous updates as new content is predicted relevant to the query. They showed that 96% of information filtering queries (and 73% of ad-hoc queries) were a combination of *context queries*, which aim to track mentions of a named entity (e.g., "Microsoft"), and *concept queries*, which seek posts about a general topic (e.g., "stock trading").

Given that the inherent nature of feed retrieval is similar to that of filtering, queries are expected to be more general and multifaceted than queries observed in ad-hoc search. The topics relevant to feed search should be capable of stimulating discussion in a particular blog over an extended period of time. Of the topics used in the TREC 2007 Blog Distillation Task (Macdonald, Ounis, & Soboroff 2007) and in this evaluation, 12/45 involved a named entity (e.g., "Nintendo DS", "violence in Sudan", "Solaris"). The majority, 33/45, were concept queries (e.g., "tennis", "home baking", "photography"). Thus, we can then view the goal of AQE for blog search as broadening the scope of a concept query with terms that relate to the different aspects or dimensions of the topic, in addition to simply enhancing the query with synonymous terms.

We investigate two methods of query expansion in this work. The first method, Indri's built-in pseudo-relevant feedback (PRF) using the blog corpus, is a strong baseline borrowed from ad-hoc retrieval. The second, our Wikipedia-based technique, we expect to be robust against splog content by pulling expansion terms from an external, possibly cleaner, corpus. By focusing on anchor text pointing to likely relevant or related pages from Wikipedia, this expansion technique also attempts to capture phrases that characterize widely different aspects of the topic.

Target Corpus Pseudo-Relevance Feedback

A typical method of performing automatic query expansion is pseudo-relevance feedback. This method assumes the top retrieved documents are relevant, identifies terms from within those documents that distinguish them from the collection, and adds those terms back to the query. Indri's built-in pseudo-relevance feedback mechanism is based on Lavrenko's relevance model (Lavrenko & Croft 2001). In this model, a language model is built from the top retrieved documents, and terms from that language model are ranked by their weights. A weighted unigram query is then built with those top terms, and this query is combined with the original query to retrieve the final set of documents. Previous results using this technique show strong performance in ad-hoc retrieval (Metzler *et al.* 2006; Diaz & Metzler 2006).

We experimented with several different settings of the pseudo-relevance feedback parameters, and the best results are reported below. These results correspond to building the relevance model with the top 10 retrieved documents, adding 20 feedback terms to the query and using a weight of 0.6 on the original query and 0.4 on the expanded query. The relevance model is built from the top retrieved feeds in the large document model and the top retrieved entries in the small document models. These parameter settings are in the range of what has been effective for other retrieval tasks. Although more training data is necessary to effectively tune all the parameters used in Indri's pseudo-relevance feedback model, we believe the results reported below are an accurate representation of the effectiveness of this model on the Blog06 corpus.

Wikipedia-based Expansion

Our simple Wikipedia-based expansion technique was motivated by the observation that valuable expansion ngrams (i.e., phrases) are realized in the anchor text of hyperlinks pointing to Wikipedia articles that are relevant to the base query.

Wikipedia Preprocessing Wikipedia articles are available for download in their original markup language, which encodes useful metadata such as the article's title and its outgoing hyperlinks, consisting of the title of the target page and an optional anchor phrase. When the author specifies an anchor phrase, this phrase can be considered synonymous with the hyperlink target's title (e.g., "US$" $\rightarrow$ "United States dollar"). About 2.4 million articles from the English Wikipedia were indexed using the Indri search engine. The article's title and hyperlinks (anchor text and target page title) were indexed as structural elements.

Algorithm Our Wikipedia-based expansion algorithm proceeds as follows. First, the base query is run as a dependence model query on the Wikipedia corpus. The result is a ranked list of Wikipedia articles, in descending order of predicted relevance. From this ranked list, we define the top-ranked R documents as the *relevant set*, $S_{\mathrm{R}} = \{d_0, d_1, ..., d_{\mathrm{R}}\}$, and the top-ranked W documents as the *working set*, $S_{\mathrm{W}} = \{d_0, d_1, ..., d_{\mathrm{W}}\}$, where $\mathrm{R} \leq \mathrm{W}$. Note that $S_{\mathrm{R}} \subseteq S_{\mathrm{W}}$. Then, each anchor phrase a_i appearing in a document in S_{W} and linking to a document in S_{R} is scored according to

[4] http://www.blogdigger.com/index.html

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

where a_i is a unique anchor phrase and a_{i_j} is an occurrence of anchor phrase a_i. The function $target(a_{i_j})$ returns the target article linked to by occurrence j of anchor phrase a_i. The function $rank(\cdot)$ returns the rank of the Wikipedia article in the ranked list returned in response to the base query. $\mathbb{I}(\cdot)$ is the identity function, which equals 1 if its argument is true and 0 otherwise. The score of anchor phrase a_i, $score(a_i)$, is greater when many occurrences of a_i appear in hyperlinks within S_{W} that link to a highly ranked page within S_{R}. In our experiments $\text{R} = 500$ and $\text{W} = 1000$ and were selected ad hoc. Anchor text appearing less than 3 times was ignored and the most highly scoring 20 anchor text phrases were chosen as expansion phrases for the base query. The scores of the top 20 expansion terms were normalized to sum to 1. Each expansion phrase's normalized score was used to weight it with respect to the other 19 expansion phrases in the resulting `#weight` query submitted to Indri. The final retrieval gives the expanded query and the original query equal weights.

Intuitively, R and W play different roles. W limits the size of the search space from where expansion phrases are pulled. R controls the range of topical aspects that candidate expansion terms may cover. A large value of W in combination with a small value of R biases the algorithm towards finding phrases synonymous with the few concepts assumed to be most relevant to the query. Increasing the value of R increases the range of topics represented by candidate expansion terms since the algorithm assumes that a greater number of top-ranked Wikipedia articles are relevant to the query.

One natural question is how sensitive this method is to parameter R, the size of the relevant set S_{R}, and W, the size of the working set S_{W}. Ultimately, only anchor phrases linking to a document in S_{R} are considered, irrespective of the rank of the document containing the anchor text. W should be large enough (≥ 1000) to allow enough anchor phrases to compete. More interesting is the effect of R on the top-ranked T expansion phrases.

To test this, we conducted an experiment in which $W = 1000$ was held constant and R was increased from $R^* = 50$ to $R^* = 800$ in increments of 50. We performed these experiments with the queries from the TREC 2007 Blog Distillation Task, explained in more detail below. At each increment, we computed, for each query, the percent overlap between the top T expansion terms chosen with that value of R^* and the top T expansion terms chosen with a fixed $R = 50$. Figure 1 shows the average percent overlap between the top T terms (i.e., $T = 1, 5, 10, 20$) chosen for a value of R^* and those chosen with $R = 50$. For example, when $R^* = 500$, on average, about 60% of the top 5 terms ($T = 5$) were the same as those selected with $R = 50$. When $R^* \geq 250$, on average, the top expansion term ($T = 1$) was the same as that chosen with $R = 50$ about 87% of the time. Figure 1 conveys two major points. First, the algorithm is less sensitive to R for larger values of R (all curves flatten as R increases). Second, the top 1 to 5 expansion terms are more stable than the top 10 to 20 expansion terms for different values of R. Of the weighted query expansion terms returned by the algorithm, the expansion terms that get replaced by varying R are the least influential ones, not the most heavily weighted terms from the top of the list. The most heavily weighted terms are fairly stable. This is a desireable property.

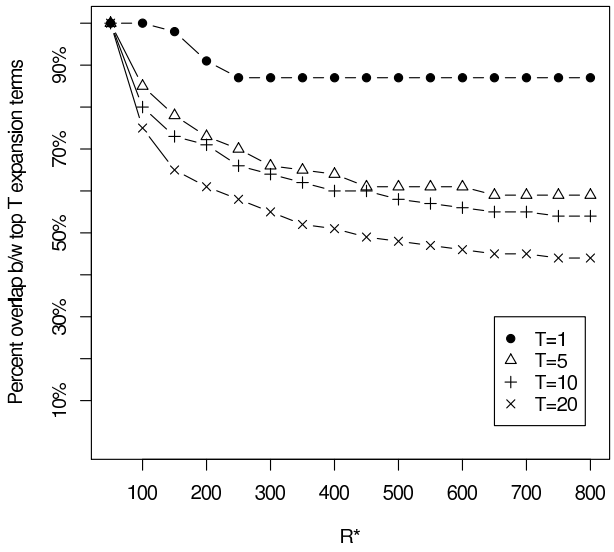

Figure 1: Average percent overlap between the top T terms ($T = 1, 5, 10, 20$) with R^* = 50,100,...,800 and the top T terms with R = 50.

Data

Our experiments were conducted using the TREC Blog06 collection, which was used in all tasks in the TREC 2006 and TREC 2007 Blog Track (Macdonald, Ounis, & Soboroff 2007). The collection represents a large sample of the blogosphere crawled over an eleven week period, from December 6, 2005 to February 21, 2006. The collection size is 148GB and has three main components: (1) 100,649 feeds, (2) over 3.2 million permalinks, and (3) 324,800 homepage documents. Feed sizes range from under ten posts in this time frame to over ten thousand. To remain representative of the blogosphere, the collection contains a significant portion of spam, non-English blogs, and non-blog content. More details about the Blog06 collection and how it was created are found in (Macdonald & Ounis 2006).

We chose to focus exclusively on feeds and ignore permalinks and homepage documents. Feed documents were a combination of ATOM and RSS XML. These two formats contain different XML elements that we mapped into a unified representation in order to make use of the structural elements within the feed (i.e., the feed title, the feed description, the entry title, and the entry text). During corpus collection, content was pulled from each feed weekly, which means that duplicate posts in some cases appear in consecutive feed fetches. Duplicate posts were removed

to avoid biasing towards the content of these posts. Documents were stemmed using the Krovetz stemmer and common stop words were removed as well as manually identified web- and feed-specific stop words, such as "www", "html", and "wordpress". Feeds with fewer than 4 posts and non-English feeds (as specified in either the `feed.lang` or `channel.language` element) were ignored.

The 45 queries and their associated relevance judgements used in this evaluation were exactly those used in the Blog Distillation Task at TREC 2007. These 45 queries were selected by the task's organizers from a set of queries proposed by the task's participants, who proposed topics based on what they would envision submitting to a feed retrieval system.

Experimental Results & Analysis

We evaluated 18 different methods of feed retrieval, by combining 6 different representations with and without two query expansion methods (i.e., $18 = 6 \times 3$).

Document Representations

The large document representation, `LD`, retrieves feed documents based on equation 1. The baseline small document model, `SD`, aggregates posts into a feed ranking according to equation 2 using the equation 3 to estimate $P(E|F)$. `+SM` indicates the small document smoothed estimates of $P(E|F)$ according to equation 4. `SD+Int` is the baseline small document model with the interpolated language model in equation 5. `SD+Int+SM` uses both equations 5 and 4. Finally, the `Combined` model linearly combines the best small and large document models (`SD+Int+SM` and `LD`) with equal weights as follows.

$$Rel(F) = 0.5 \times P(Q|F) + 0.5 \times \sum_{E \in F} P(Q|E)P(E|F)$$

Table 1 shows results in terms of mean average precision (MAP) and precision at 10 (P@10) for all 6 document representations without any query expansion. Significance testing was done using the Wilcoxon signed rank test. The methods are ordered from top to bottom in ascending order of complexity. In Table 1, a statistically significant improvement over the simpler models is shown with $(\star, \dagger, +)$.

None of the (non-expanded) retrieval models show a statistically significant improvement in terms P@10. In terms of MAP, both enhancements to the baseline small-document model (`+SM` and `+Int`) result in a statistically significant improvement. A further significant improvement is realized by combining these two enhancements in `SD+Int+SM`. The best small-document model, `SD+Int+SM`, and the large document model, `LD`, are statistically indistinguishable. The two models, however, appear to be complementary as the combined performance is greater than either model on its own.

Model	MAP	P@10
LD	0.290 $\star$ † †+	**0.400**
SD	0.201	0.353
SD+SM	0.245 $\star$	0.371
SD+Int	0.267 $\star$	0.389
SD+Int+SM	0.286 $\star$ † +	**0.400**
Combined	**0.293** $\star$ † +	0.398

Table 1: MAP and P@10 for small document models (without query expansion). $\star$ indicates significant improvement ($p \leq 0.005$) over the `SD` model. †/ † † and +/ + + indicate significant improvement over the `SD+SM` and `SD+Int` models respectively at the $p \leq 0.05/p \leq 0.01$ levels.

Query Expansion

Table 2 shows results in terms of mean average precision (MAP) and precision at 10 (P@10) for all 6 document representations with and without the two query expansion methods, Indri's built-in pseudo-relevance feedback (**+P**) and our Wikipedia-based expansion (**+W**). A statistically significant improvement over the baseline, unexpanded queries, is marked with $(\dagger, \star)$.

Based on these results, Indri's pseudo-relevance feedback did not provide any noticeable performance improvement for any document representation model and in many cases slightly degrades retrieval performance. We believe this is because the Blog06 corpus, and blog data in general, is extremely noisy and poorly suited for use as a source for high-quality terms to add into a query. Wikipedia-based expansion, in contrast, shows consistent improvement across all retrieval models in terms of both MAP and P@10. This improvement is statistically significant in terms of MAP in all cases and in terms of P@10 in four of the six retrieval models. Typically, when query expansion techniques work well, they improve performance on retrieval measures with an emphasis on document recall, like MAP. Rarely is a consistent improvement shown for high-precision evaluation measures such as P@10. Significantly improving precision after query expansion (via any method of query expansion) is a positive result.

Wikipedia-based expansion is independent of the document representation adopted for feed search since it mines expansion phrases from an external corpus, the Wikipedia, using a fixed representation. The same 20 expansion phrases were used to expand each original query under all 6 document representations we evaluated. In this error-analysis we focus on the effect of Wikipedia-expansion on the large document approach, `LD`, our simplest representation. In terms of MAP, Wikipedia-based expansion improved results for 39 out of 45 queries (about 87%). Table 3 shows the top 10 expansion phrases selected for the two queries helped the most by Wikipedia-based expansion, "US election 2008" and "home baking". Wikipedia-based expansion hurt performance for 6/45 queries (about 13%). One disadvantage of our Wikipedia-based expansion technique is that parameter R, the size of the relevant set S_{R}, is query independent. R determines the number of top ranked Wikipedia articles that are assumed relevant to the query. For some of the 6/45 queries, the top few ranked articles returned in response to the base query were relevant, but the articles quickly drifted off topic down the ranking. Two cases of this

Model	MAP		
	–	+P	+W
LD	0.290	0.283	0.356⋆
SD	0.201	0.203	0.236⋆
SD+SM	0.245	0.245	0.279⋆
SD+Int	0.267	0.273	0.346⋆
SD+Int+SM	0.286	0.269	0.342⋆
Combined	0.293	0.285	**0.357**⋆
	P@10		
LD	0.400	0.391	0.473†
SD	0.353	0.347	0.378
SD+SM	0.371	0.353	0.400
SD+Int	0.389	0.397	**0.480**⋆
SD+Int+SM	0.400	0.379	0.458†
Combined	0.398	0.400	0.473†

Table 2: MAP and P@10 for all models with Wikipedia (+W), with Indri's Pseudo-relevance feedback (+P), and with neither (–). Statistical significance over the baseline (no feedback) model is indicated by † $= (p \leq 0.01)$ and ⋆ $= (p \leq 0.001)$.

republican	bread
John McCain	baking
U boat	flour
Al Gore	butter
Republican Party	baking powder
Barack Obama	cake
Rudy Giuliani	yeast
2008 presidential election	wheat
republicans	food
Ron Paul	cookie

Table 3: Top 10 (out of 20) query expansion phrases selected for TREC query 991, "US election 2008" (left), and TREC query 967, "home baking" (right). The percent improvement in terms of MAP of LD with vs. without Wikipedia-based expansion was 3184% and 716%, respectively.

were the queries "Apple iPod" and "Google Maps Earth". In the case of "Apple iPod", the top few Wikipedia articles were relevant (i.e., "iPod", "iPod Shuffle", "iTunes", "iPod Mini"), but down the rank, the topic quickly drifted into the more general topic of "Apple", with articles such as "Apple Inc", "history of Apple", "typography of Apple", "Apple Inc. advertising". The same was the case for query "Google Maps Earth". The top few articles were relevant (i.e., "Google Maps", "Google Earth", "web mapping"), but the topic quickly drifted into the more general topic of "Google" (i.e., "Google", "censorship by Google", "Google Groups"). Table 4 shows the top 10 expansion phrases selected for the two queries hurt the most by Wikipedia-based expansion, "stock trading" and "vacation travel".

New York Stock Exchange	vacation
stock exchange	permanent vacation
stock	Chevy Chase
London Stock Exchange	Aerosmith
insider trading	tourism
finance	time travel
options	Vegas vacation
bonds	Walt Disney World Resort
corporation	time machine
bond	John Hughes

Table 4: Top 10 (out of 20) query expansion phrases selected for TREC query 986, "stock trading" (left), and TREC query 995, "vacation travel" (right). The percent improvement in terms of MAP of LD with vs. without Wikipedia-based expansion was -33% and -27%, respectively.

Related Work

Retrieval Models for Feed Search

Prior work in retrieval models for feed search has drawn analogies between feed retrieval and several other retrieval tasks: cluster-based retrieval (Seo & Croft 2007), resource selection in distributed information retrieval (Elsas *et al.* 2007), and expert finding (Hannah *et al.* 2007). All of these tasks share the common goal of ranking units of retrieval that are defined as collections of documents.

(Seo & Croft 2007) approached feed retrieval as a special case of cluster-based retrieval. Cluster-based retrieval attempts to pre-process the document corpus into topical clusters, rank those clusters in response to a query, and finally retrieve documents from the most highly ranked clusters. Their algorithm treats each feed as a pseudo-cluster of blog posts. Posts are retrieved in an intermediate step according to $P(Q|D)$ and aggregated into a final ranking of feeds according to

$$P(Q|F_i) = \left(\prod_{j=1}^{K} P(Q|D_j) \right)^{\frac{1}{K}},$$

where the product is over the K most highly ranked posts, D_j, belonging to feed F_i (i.e., $D_j \in F_i$). For feeds with fewer than K retrieved posts, the product above is padded using "phantom" posts, each assumed to have a query likelihood of $P_{min}(Q) = \min_{D_k \in C} P(Q|D_k)$, the minimum query likelihood from the entire set of posts retrieved. Seo and Croft experimented with two representations in isolation and in combination: a pseudo-cluster representation, described above, and a global representation, where blog feed documents are created by concatenating the constituent posts and retrieved according to $P(Q|F)$. Their global representation outperformed their pseudo-cluster representation. However, an improvement over both representations was obtained by combining the two.

(Elsas *et al.* 2007) approached feed retrieval as a special case of resource ranking in distributed information retrieval, by adapting the existing ReDDE algorithm (Si & Callan 2003). An intermediate ranking of posts is aggre-

gated into ranking of feeds according to

$$P(Q|F_i) = \sum_{D_j \in \hat{F}_i} P(Q|D_j).$$

To normalize for feed size (measured in number of posts) an index was created by randomly sampling (with replacement) 100 posts from each feed. The sum above is over all entries belonging to $\hat{F}_i$, a sample of 100 posts randomly selected from F_i. This was done to prevent less-relevant large feeds from unjustifiably ranking better than more-relevant smaller feeds. As in (Seo & Croft 2007), a baseline global (large document) representation also outperformed the federated (small document) model.

(Hannah *et al.* 2007) approach feed retrieval as a special case of expert finding, which is the task of ranking potential experts in an subject described by a user's query. Candidate experts are usually represented by their email correspondence and the major assumption is that an expert will have a large volume of email that is relevant to the query. As in feed retrieval, the unit of retrieval (the candidate expert) is a collection of documents (their email correspondence). (Hannah *et al.* 2007) approached the problem of feed search by aggregating posts into a blog ranking based on two criteria: relevance and cohesiveness. Relevance is analogous to $P(Q|D)$ in the language model-based approaches adopted by (Seo & Croft 2007) and (Elsas *et al.* 2007). Cohesiveness is a query-independent property of a blog that measures the divergence in language across its constituent entries. While ranking, their algorithm favors blogs more centered on a single topic.

Query Expansion Models for Feed Search

Our query expansion approach builds upon prior work on using external collections as a source of query expansion terms. (Diaz & Metzler 2006) extended PRF with Lavrenko's relevance models (Lavrenko & Croft 2001), our baseline query expansion method, to handle multiple external corpora. A nearly consistent improvement was observed when the external corpus was a superset of the target corpus. Interestingly, an improvement was also observed in some cases where the target corpus contained a different type of document than the external corpus (e.g., news vs. web documents). This agrees with our findings that the target and external corpora need not share the same document type (e.g., informal blogs vs. well-edited encyclopedia articles). (Diaz & Metzler 2006) show that the topic coverage of the external corpus matters more than its size and that a large external corpus is neither necessary nor sufficient. However, using the web (a very large corpus) as a source of expansion terms has been effective in handling difficult queries, with possibly few relevant documents in the target corpus (Voorhees 2004; 2005).

(Y. Li & Chung 2007) also used the Wikipedia for query expansion, using the category assignments of Wikipedia articles. The base query is run against the Wikipedia and each category is assigned a weight proportional to the number of top-ranked articles assigned to it. Articles are then re-ranked based on the sum of the weights of the categories to which each belongs. From this final article ranking, expansion terms are selected from the top documents. Our Wikipedia-based algorithm differs from (Y. Li & Chung 2007) in that we focus on anchor text and that our candidate phrase scoring function combines relevance with centrality. An anchor phrase used in links to top ranked articles can be outscored by one used in links to articles ranked lower, if the latter is more frequent.

Conclusion

We developed two different and equally effective document representation techniques for blog retrieval: the large document model (`LD`) and the smoothed small document model (`SD+Int+SM`). These results show that although a small-document model may be adapted for many of the unique aspects of blog retrieval, care must be taken in properly representing blogs as individual posts and in aggregating a post ranking into a blog ranking. Nonetheless, both of these retrieval models are effective and they appear to be complementary – combining the two models provides superior results to either model in isolation. Our comparison of these different document representations for blog recommendation suggests several promising research directions. First, a deeper understanding of the situations in which one representation may be preferable to another could yield even further improvements. Although the two models achieve comparable performance on average, there are some queries in which one model greatly outperforms the other. Secondly, considering variable entry normalization schemes based on some measure of "centrality" of the entry to the blog (eq. 2) is another promising research direction. Intuitively, we would like to favor a post that is "close to" its blog's central topic, rather than an occasional post on the topic under consideration.

In addition to evaluating several document representation models, we presented a simple and novel algorithm for using anchor text from Wikipedia to expand users' queries. This expansion model yielded consistent and significant improvements with both high-recall *and* high-precision evaluation measures under all document representations. In contrast, pseudo-relevance feedback, a technique popular in ad-hoc search did not improve results. Further refinements of the Wikipedia model could include automatically learning when not to expand a query or learning a query-dependent cutoff for the relevant set size $|S_{\mathrm{R}}|$, the top Wikipedia articles assumed to be relevant to the query. In this work, we did not make use of the rich metadata associated with Wikipedia articles such as category labels or of the greater link-network structure of a document within Wikipedia. Both of these features could help refine our concept of the relevant document set to consider for mining expansion phrase candidates.

Acknowledgements

We wish to thank the anonymous reviewers for their valuable comments. This work was supported by the eRulemaking project and NSF grant IIS-0240334 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.

References

Callan, J. 2000. *Distributed information retrieval*. Kluwer Academic Publishers.

Diaz, F., and Metzler, D. 2006. Improving the estimation of relevance models using large external corpora. In *Proceedings of SIGIR 2006*.

Elsas, J.; Arguello, J.; Callan, J.; and Carbonell, J. 2007. Retrieval and feedback models for blog distillation. In *Proceedings of TREC 2007*.

Hannah, D.; Macdonald, C.; Peng, J.; He, B.; and Ounis, I. 2007. University of glasgow at TREC 2007: Experiments with blog and enterprise tracks with terrier. In *Proceedings of TREC 2007*.

Java, A.; Kolari, P.; Finin, T.; Joshi, A.; and Oates, T. 2007. Feeds that matter: A study of bloglines subscriptions. In *Proceedings of ICWSM 2007*.

Kolari, P.; Java, A.; and Finin, T. 2006. Characterizing the splogosphere. In *Proceedings of WWW 2006*.

Lavrenko, V., and Croft, W. B. 2001. Relevance based language models. In *Proceedings of SIGIR 2001*.

Lin, Y.-R.; Sundaram, H.; Chi, Y.; Tatemura, J.; and Tseng, B. L. 2006. Splog detection using self-similarity analysis on blog temporal dynamics. In *Proceedings of AIRWeb 2006*.

Macdonald, C., and Ounis, I. 2006. The TREC Blog06 collection: Creating and analysing a blog test collection. *DCS Technical Report TR-2006-224*. Dept. of Computing Science, Univ. of Glasgow.

Macdonald, C.; Ounis, I.; and Soboroff, I. 2007. Overview of the TREC 2007 blog track. In *Proceedings of TREC 2007*.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Metzler, D., and Croft, B. W. 2004. Combining the language model and inference network approaches to retrieval. *Information Processing and Management* 40(5): 735–750.

Metzler, D., and Croft, B. W. 2005. A markov random field model for term dependencies. In *Proceedings of SIGIR 2005*.

Metzler, D.; Strohman, T.; Zhou, Y.; and Croft, B. 2006. Indri at TREC 2005: Terabyte track. In *Proceedings of TREC 2005*.

Mishne, G., and de Rijke, M. 2006. A study of blog search. In *Proceedings of ECIR 2006*.

Mishne, G. 2007. Using blog properties to improve retrieval. In *Proceedings of ICWSM 2007*.

Ogilvie, P., and Callan, J. 2004. Hierarchical language models for retrieval of xml components. In *Proceedings of INEX 2004*.

Seo, J., and Croft, W. B. 2007. Umass at TREC 2007 blog distillation task. In *Proceedings of TREC 2007*.

Si, L., and Callan, J. 2003. Relevant document distribution estimation method for resource selection. In *Proceedings of SIGIR 2003*.

Voorhees, E. 2004. Overview of the TREC 2004 robust retrieval track. In *Proceedings of TREC 2004*.

Voorhees, E. 2005. Overview of the TREC 2005 robust retrieval track. In *Proceedings of TREC 2005*.

Y. Li, R.W.P. Luk, E. H., and Chung, F. 2007. Improving weak ad-hoc queries using wikipedia as external corpus. In *Proceedings of SIGIR 2007*.