# Document and Query Expansion Models for Blog Distillation

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

## Abstract

This paper presents the CMU submission to the 2008 TREC blog distillation track. Similar to last year's experiments, we evaluate different retrieval models and apply a query expansion method that leverages the link structure in Wikipedia. We also explore using a corpus that combines several different representations of the documents, using both the feed XML and permalink HTML, and apply initial experiments with spam filtering.

## 1 Introduction

The CMU submission to the 2008 blog distillation track explored document representation, retrieval models, query expansion, and spam filtering. CMU's retrieval system, based on the Indri search engine [1], used a combined index of the permalink and feed documents, differentially weighting text from various parts of the HTML and XML. Two retrieval models were applied: the *large document model*, where each feed is viewed as a single document; and the *small document model*, where a feed is represented as a collection of individual entry documents. Similarly to last year's submission, our query expansion method leverages the link structure in Wikipedia. A spam filtering component was also integrated.

## 2 Document Representation

Although our system last year successfully made use of only the feed (XML) documents, subsequent testing indicated that using the permalink (HTML) documents could provide some performance improvements. In order to leverage both representations of the blog feeds, this year's submission used a combined index in which each indexed blog contains the text from both the permalink and feed documents. These separate representations of the blogs are indexed as fields in Indri, allowing flexible access to the different document representations at query time.

The fields represented in our index, given in Table 1, include the full text of the HTML pages (`permtext`), as well as several structural elements of the feed XML documents.

| Field Name | Description (Source) |
|---|---|
| `permtext` | Permalink Text (HTML) |
| `title` | Feed title (XML) |
| `entrytitle` | Entry title (XML) |
| `entrybody` | Entry Content (XML) |

Table 1: Indexed fields

## 3 Retrieval Models

We applied two retrieval models to the task of blog distillation: the large document model, which treats each blog or feed as a single (large) document, and the small document model, which retrieves blog entries individually and aggregates an entry ranking into an overall feed ranking.

### 3.1 Large Document Model

The large document model treats each feed as a concatenation of all its entries. In this model, documents are ranked by the posterior probability of observing the feed given the query,

$$\begin{aligned} P_{LD}(F|Q) &= P_{LD}(Q,F)/P(Q) \\ &\overset{rank}{=} \underbrace{P(F)}_{\substack{\text{Document}\\\text{Prior}}} \underbrace{P_{LD}(Q|F)}_{\substack{\text{Query}\\\text{Likelihood}}}. \end{aligned} \quad (1)$$

The query likelihood component is estimated as a weighted combination of query likelihoods from the different document representations: permalink text, feed title, entry titles and entry content,

$$P_{LD}(Q|F) = \prod_j P_{LD}(Q|F^{(j)})^{v_j}, \quad (2)$$

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

where the $j$ denotes different representations and $v_j$ are learned weights for each of those representations. This representation-specific query likelihood component is estimated with Metzler & Croft's *full dependence model* [6, 2] using Dirichlet-smoothed maximum likelihood estimates [7],

$$\begin{aligned} P_{LD}(Q|F^{(j)}) &= \prod_{\psi_i \in \Psi(Q)} P_{LD}(\psi_i|F^{(j)})^{w_i} \\ &= \prod_{\psi_i \in \Psi(Q)} \left( \frac{tf_{\psi_i;F^{(j)}} + \mu P_{MLE}(\psi_i|C)}{|F^{(j)}| + \mu} \right)^{w_i} . \end{aligned} \tag{3}$$

The $\psi_i$ are query unigram and term window query features, and the dependence model weights $w_i$ are taken from previous literature. This complex query formulation can be expressed in the following Indri query template shown in Table 2, where `<unigrams>`

```
#weight( v1 #weight(
    0.8 #combine(<unigrams>.(permtext))
    0.1 #combine(<ordered windows>.(permtext))
    0.1 #combine(<unordered windows>.(permtext)))
 v2 #weight(
    0.8 #combine(<unigrams>.(title))
    0.1 #combine(<ordered windows>.(title))
    0.1 #combine(<unordered windows>.(title)))
 v3 #weight(
    0.8 #combine(<unigrams>.(entrytitle))
    0.1 #combine(<ordered windows>.(entrytitle))
    0.1 #combine(<unordered windows>.(entrytitle)))
 v4 #weight(
    0.8 #combine(<unigrams>.(entrybody))
    0.1 #combine(<ordered windows>.(entrybody))
    0.1 #combine(<unordered windows>.(entrybody))))
```

Table 2: Large Document Indri Query Template

is a simple unigram query, `<unordered windows>` is a group of `#uw` query operators, each with a window size set to twice the number of query terms considered, and `<ordered windows>` is a group of `#1` query operators. Parameters `v1`-`v4` were trained on last year's queries and will be discussed in Section 3.3.

## 3.2 Small Document Model

The small document model scores each entry individually, and then combines those scores into an overall feed score. Again, ranking by the posterior probability of observing the feed given the query, we have,

$$\begin{aligned} P_{SD}(F|Q) &= \frac{1}{P(Q)} \sum_{E \in F} P_{SD}(Q,E,F) \\ &\overset{rank}{=} P(F) \sum_{E \in F} P(Q|E,F) P(E|F) \\ &\overset{rank}{=} \underbrace{P(F)}_{\substack{\text{Feed} \\ \text{Prior}}} \sum_{E \in F} \underbrace{P(Q|E)}_{\substack{\text{Query} \\ \text{Likelihood}}} \underbrace{P(E|F)}_{\substack{\text{Entry} \\ \text{Centrality}}} , \end{aligned} \tag{4}$$

where the last line holds if we assume queries are conditionally independent of feeds given the entry. As in the large document model, Equation 2, the query likelihood is calculated via a combination of different document representations

$$P_{SD}(Q|E) \quad = \quad \prod_j P_{SD}(Q|E^{(j)})^{v_j}, \tag{5}$$

and these query likelihood components are estimated with a full dependence model using Jelinek-Mercer smoothing [6, 7] to combine both the entry and feed language models,

$$\begin{aligned} P_{SD}(Q|E^{(j)}) &= \prod_{\psi_i \in \Psi(Q)} P_{JM}(\psi_i|E^{(j)})^{w_i} \\ &= \prod_{\psi_i \in \Psi(Q)} \Big( \lambda_E P_{MLE}(\psi_i|E^{(j)}) \\ &\qquad + \lambda_F P_{MLE}(\psi_i|F^{(j)}) \\ &\qquad + \lambda_C P_{MLE}(\psi_i|C) \Big)^{w_i} \end{aligned} \tag{6}$$

The centrality component of this model is given by,

$$P(E|F) = \frac{\phi(E,F)}{\sum_{E_i \in F} \phi(E_i,F)}. \tag{7}$$

where $\phi$ is defined as,

$$\phi(E,F) = \left( \prod_{t_i \in E} P(t_i|F)^{\frac{tf_{t_i;E}}{|E|}} \right). \tag{8}$$

This centrality scoring favors entries that share a language more closely with the language of the feed as a whole. In practice, the product in Equation 8 is only taken over the query terms, providing a query-conditioned centrality measure. The feed prior component of this model is used to correct for the overly optimistic centrality scoring, and is proportional to the log of the feed length: $P(F) \propto log(|F|)$. See [2] for a more thorough description of this retrieval model.

### 3.3 Parameter Estimation

The above models have several free parameters for smoothing and model combination that must be set. All of these parameters were set via a simple grid search with a step size of 0.1 using last year's queries and relevance judgements [5]. The parameter settings were used in Equations 5 and 2 are shown in Table 3.

| Model | `permtext` (`v1`) | `title` (`v2`) | `entrytitle` (`v3`) | `entrybody` (`v4`) |
|---|---|---|---|---|
| LD | 0.3 | 0.5 | 0.1 | 0.1 |
| SD | 0.6 | – | 0.1 | 0.3 |

Table 3: Weight settings ($v_i$) for different document representations.

The smoothing parameters used were $\mu = 2500$ for the large document model and $\lambda_E = 0.4, \lambda_F = 0.3$ and $\lambda_C = 0.3$ for the small document model.

## 4 Wikipedia Link-based Expansion

As demonstrated at last year's TREC submission, blog retrieval can greatly benefit from specifically designed query expansion methods. In this year's submission, we apply the same query expansion method that utilizes the link structure in Wikipedia to discover topics related to the query.

The original Wikipedia markup includes cross-article links. Each link is specified by its target Wikipedia article and its anchor text or *anchor phrase*, which may differ from the target article's title. Our Wikipedia link-based expansion method expands the query with related anchor phrases from Wikipedia, scoring each proportional to how often it occurs in links to documents relevant to the query.

The unexpanded query is issued as a dependence model query to our Wikipedia index, comprised of $2,471,311$ articles, excluding date and category pages, from the English Wikipedia. From the resulting ranking, two document sets are defined: the top $R$ documents are defined as the *relevant set*, $\mathcal{S}_R$, and the top $W$ documents are defined as the *working set*, $\mathcal{S}_W$. Because $R < W$, it follows that $\mathcal{S}_R \subset \mathcal{S}_W$. The method focuses on anchor phrases appearing in articles in $\mathcal{S}_W$ that link to an article in $\mathcal{S}_R$. Anchor phrase $a_i$ is scored according to,

$$\lambda_i = \sum_{a_{i_j} \in \mathcal{S}_W} \mathcal{I}\Big(\text{target}(a_{i_j}) \in \mathcal{S}_R\Big) \times \Big(R - \text{rank}\big(\text{target}(a_{i_j})\big)\Big), \qquad (9)$$

where $a_{i_j}$ denotes an occurrence of anchor phrase $a_i$, $\text{target}(a_{i_j})$ is the article linked to by anchor phrase occurrence $a_{i_j}$, $\text{rank}\big(\text{target}(a_{i_j})\big)$ denotes the rank of $\text{target}(a_{i_j})$, and $\mathcal{I}$ is the identity function. The unexpanded query was augmented with the most highly scoring 20 expansion phrases.

The method aims to fulfill two desired properties of expansion phrases — that they relate to the query and that they are popular terms, likely to appear in other documents. If two candidate expansion phrases appear equal number of times in $\mathcal{S}_W$, the one appearing in links to the most highly ranked documents will score higher. If two candidate expansion terms appear in links to the same document, the most frequent one will score higher. Prior work shows this method resulting in higher retrieval performance than pseudo-relevance feedback (PRF) on the same external resource, the Wikipedia [2]. An evaluation of the sensitivity of this query expansion method to the parameters $R$ and $W$ is presented in previous work [2, 1], and the parameter settings used here are taken from that work.

## 5 Splog Detection

Our method for splog detection combined four distinct classifiers. Three were rule-based classifiers based on potentially high precision, low coverage heuristics. The fourth classifier is a maximum-margin classifier based on bag-of-word features.

### 5.1 Post Time Interval

Some splogs are machine-generated, containing nonsense text or even snippets of text weaved together from other blogs or websites [4]. Machine-generated splogs may be characterized by an unusually consistent time interval between consecutive posts. Our post time-interval classifier (TI) classifies a blog as spam if its time interval between consecutive posts varies by no more than 10 seconds. This classifier labeled 366 blogs as spam, of which 333 (91%) were not labeled as spam by any other classifier.

### 5.2 Term Compression

Many splogs exist to promote and advertise affiliated sites [3]. These splogs may be characterized by an unusually small ratio of vocabulary size to term count. Our term compression classifier (TC) classifies a blog as spam if $X \leq 6.5\%$ of its unique terms account for $Y \geq 50\%$ of its total term count. We set $Y = 50\%$ and tuned $X$ to maximize MAP on last year's queries and relevance judgements [5]. This classifier labeled

4523 blogs as spam, of which 1687 (41%) were not labeled as spam by any other classifier.

### 5.3 Link Compression

Some splogs exist to artificially inflate the PageRank of affiliated sites [3]. These splogs may be characterized by an unusually high percentage of hyperlinks linking to the same URL(s). Our link compression classifier (`LC`) classifies a blog as spam if $X \leq 1\%$ of its unique link-to URLs account for $Y \geq 70\%$ of its total hyperlink count. We set $Y = 70\%$ and tuned $X$ to maximize MAP on last year's queries and relevance judgements [5]. This classifier labeled 640 blogs as spam, of which 602 (94%) were not labeled as spam by any other classifier.

### 5.4 SVM classifier

The lexical features of a blog may provide evidence that it is spam (e.g., pornographic content). An SVM bag-of-words model was trained using a publicly available ham/spam blog data set[2][3]. Terms were weighted by $P_{MLE}(w|F)$. This classifier labeled 10839 blogs as spam, of which 8365 (77%) were not labeled as spam by any other classifier.

### 5.5 Combining Classifiers

These four spam classifiers were combined log-linearly and integrated into the large document model in the form of query-independent document priors (Formula 1) as,

$$\log P(F) \propto \log P(ham|F) = \sum f_i \log \alpha_i \qquad (10)$$

where $f_i$ denotes the binary output of the above classifiers and each $\alpha_i$ denotes the weight associated with each classifier. We set $\log \alpha_{TI} = -100$ manually based on our hypothesis that its precision is high. Holding this value constant, $\log \alpha_{TC} = -3$, $\log \alpha_{LC} = -1$,and $\log \alpha_{SVM} = -1$ were set by doing a grid search to maximize MAP on last year's queries and relevance judgements [5].

## 6 Results

Our 4 submitted runs (`cmuSD` = SD model, `cmuSDWiki` = SD + Wiki expansion, `cmuLDWiki` = LD + Wiki expansion, and `cmuLDwikiSP` = LD + Wiki expansion + splog detection) used only the TREC topic title field. Results are given in Table 4.

[2]http://svmlight.joachims.org/

[3]http://ebiquity.umbc.edu/resource/html/id/212/Splog-Blog-Dataset

| run | MAP | P@10 | R-Prec |
|---|---|---|---|
| `cmuSD` | 0.246 | 0.372 | 0.3086 |
| `cmuSDwiki` | 0.259 | 0.372 | 0.3178 |
| `cmuLDwiki` | 0.302 | 0.422 | 0.3534 |
| `cmuLDwikiSP` | **0.306** | **0.434** | **0.3646** |

Table 4: Results

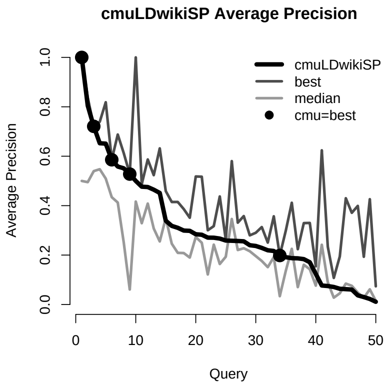
Figures 1(a) and 1(b) show `cmuLDWikiSP`'s per-query performance in terms of average precision (AP) and R-Precision (R-Prec), respectively, alongside the per-query median and best performance. Queries are sorted along the $x$ axis in descending order of `cmuLDWikiSP` performance. Dots indicate the queries for which `cmuLDWikiSP` obtained the best performance.

## 7 Wikipedia Link-based Expansion Error Analysis

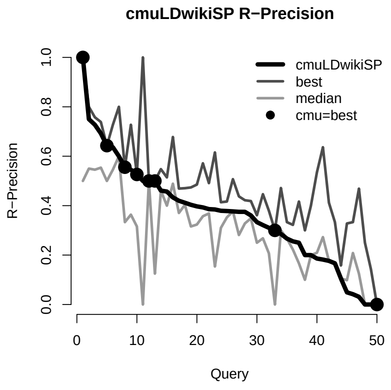
Here, we focus on the queries that were helped or hindered the most by Wikipedia link-based expansion. Comparing `cmuSD` vs. `cmuSDwiki`, Wikipedia link-based expansion improved MAP for 33/50 (66%) queries. The ones with the largest MAP increase were "road cycling" (372%), "U.S. national park" (266%) and "theater" (138%). Wikipedia link-based expansion found valuable expansion terms for these queries because most documents in the relevant set, $\mathcal{S}_R$, were in fact relevant. These were queries with many relevant articles in the Wikipedia (more than $R = 100$) in the form of relevant named entities (e.g., names of cyclists and cycling events, and names of parks and organizations dealing with parks). The top 10 expansion terms and scores for "road cycling" were,

| | |
|---|---|
| cycling | 0.135 |
| lance armstrong | 0.107 |
| uci | 0.078 |
| discovery channel pro cycling team | 0.072 |
| road bicycle racing | 0.071 |
| uci protour | 0.061 |
| paolo bettini | 0.054 |
| discovery channel | 0.051 |
| union cycliste internationale | 0.050 |
| george hincapie | 0.040 |

(33%) queries saw a decrease in MAP. One interesting observation is that non-relevant documents in $\mathcal{S}_R$ can lead to non-relevant expansion terms even when ranked below relevant documents. Wikipedia link-based expansion scores anchor phrases proportional to their frequency and the rank of the article they link to. Therefore, it is possible for an anchor phrase



(a) Per-query AP



(b) Per-query R-Prec

Figure 1: Per-query results: median, best, and `cmuSDwikiDP`

used in links to a mid-ranked article to outscore one used in links to a top-ranked article, if the first occurs more frequently than the second. In other words, scores are biased towards anchor phrases linking to (possibly non-relevant) articles with many in-links.

The query with the largest drop in MAP (70%) was "food in Singapore". The top 5 expansion terms and scores were,

| | |
|---|---|
| singapore | 0.596 |
| lee kuan yew | 0.053 |
| orchard road | 0.039 |
| national university of singapore | 0.033 |
| singapore airlines | 0.030 |

These non-relevant expansion phrases originated from links to non-relevant documents. Interestingly, the unexpanded query successfully ranked article "Cuisine in Singapore" above those linked to by these expansion phrases. However, "Cuisine in Singapore" had a total of 21 in-links. Each of the articles associated with these non-relevant anchor phrases had more than 100 in-links. Article "Singapore" had about 8000. The high number of in-links associated with these non-relevant articles in $\mathcal{S}_R$ produced these poor expansion phrases.

This same error type caused the third largest drop in MAP (30%), for the query "3D cities globes". The top 5 expansion terms and scores were,

| | |
|---|---|
| golden globe award | 0.228 |
| golden globes | 0.150 |
| duke nukem 3d | 0.084 |
| globe | 0.082 |
| golden globe | 0.066 |

Here, again, the top ranked documents in $\mathcal{S}_R$ were all relevant ("Live Search Maps", "Virtual Globe", "Google Earth", and "Polygonal Modeling"). However, all had fewer than 10 in-links. The non-relevant article "golden globe award", which contributed three non-relevant expansion phrases, was ranked 15th and had 739 in-links. The non-relevant article "Duke Nukem 3D", which contributed the third top anchor phrase, was ranked 58th and had 133 in-links.

This analysis shows that non-relevant articles in $\mathcal{S}_R$, even if not at the top of the ranking, are specially damaging when they have many in-links. These potentially damaging non-relevant articles are more likely to be introduced into $\mathcal{S}_R$ when there are less than $R$ relevant articles for the query. This analysis is compatible with prior results showing that Wikipedia link-based expansion is particularly successful when the query describes a broad, general topic, likely to have many ($> R$) relevant articles in Wikipedia [2].

## 8 Conclusion

This year we continued our evaluation of retrieval models for blog feed search, applying extensions to the previous year's models. We also experimented with query expansion for this task, as well as an expanded document representation and spam filtering. The best performing retrieval model from last year's submission continued to perform well this year, but the extensions to last year's small document model did not. Query expansion also showed promising improvements on this year's query set, and spam filtering provided a slight performance boost.

## References

[1] J. Arguello, J. Elsas, J. Callan, and J. Carbonell. Document representation and query expansion models for blog recommendation. In *ICWSM '08: Proceedings of the 2nd annual international AAAI Conference in Weblogs and Social Media*, 2008.

[2] J. L. Elsas, J. Arguello, J. Callan, and J. G. Carbonell. Retrieval and feedback models for blog feed search. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008.

[3] P. Kolari, A. Java, and T. Finin. Characterizing the splogosphere. In *WWW '06: Proceedings of the 3rd Annual Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 15th World Wide Web Conference*, 2006.

[4] Y.-R. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. L. Tseng. Splog detection using self-similarity analysis on blog temporal dynamics. In *AIRWeb '07: Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, 2007.

[5] C. Macdonald, I. Ounis, and I. Soboroff. Overview of the trec-2007 blog track. In *TREC '07: Proceedings of the 16th TREC conference*, 2007.

[6] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proc of SIGIR 05*, pages 472–479, New York, NY, USA, 2005. ACM Press.

[7] 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.