Finding people and their utterances in social media

Weerkamp, W.

Publication date
2011

5
Finding Bloggers

In the previous chapter we have looked at searching for people. Given a person name, we what are the results that characterize this person? For the most part, we have ignored the utterances of these people themselves. In this chapter we bring in the utterances and explore how we can find people by making use of what they wrote. We represent people by their utterances and use these to find the people we want to find.

The specific task we focus on in this chapter is that of *blogger finding*, or blog feed search. The goal of this task is not to return single utterances (i.e., blog posts), but to identify bloggers or blogs that show a *recurring* interest in a given topic. Bloggers who only mention the topic sporadically or in passing are considered non-relevant, but a blogger that talks about this topic regularly would be relevant. One can simply return these blogs to an end user as is, but could also decide to use the results in further processing (e.g., recommending blogs to be followed, identifying networks of expert bloggers, detecting topic shifts in blogs). Section 2.3 (page 18) gives an overview of related work in the field of blogger finding.

The total number of blogs in the world is not known exactly. Technorati,[1] the largest blog directory, was tracking 112 million blogs in 2008 and counted 175,000 new blogs every day. These bloggers created about 1.6 million entries per day and although most of these blogs are written in English, the largest part of the Internet users is not English-speaking. The China Internet Network Information Center (CNNIC)[2] released a news report in December 2007 stating that about 73 million blogs are being maintained in China, which means that, by now, the number of Chinese blogs is probably close to the number of blogs tracked by Technorati. Although we lack exact numbers on the size of the blogosphere, we can be sure that its size is significant—in terms of blogs, bloggers, and blog posts.

Given the size of the blogosphere and the growing interest in the information available in it, we need effective and efficient ways of accessing it. An important first step concerns indexing. When looking for relevant blog posts, it makes sense to do so on top of an index consisting of individual blog posts: the unit of retrieval is the same as the indexing unit, blog posts. When looking for blogs, however, two options present themselves. We could opt for the "unit of retrieval coincides with the unit of indexing" approach; this would probably entail concatenating a blog's posts into a single pseudo-

[1] http://technorati.com/blogging/feature/state-of-the-blogosphere-2008/
[2] http://www.cnnic.cn

document and indexing these pseudo-documents. In this chapter, we want to pursue an alternative strategy, viz. to drop the assumption that the unit of retrieval and the unit of indexing need to coincide for blog feed search. Instead, we want to use a post-based index (i.e., the indexing unit is a blog post) to support a blog feed search engine (i.e., the unit of retrieval is a blog). This approach has a number of advantages. First, it allows us to support a blog post search engine and a blog feed search engine with a single index. Second, result presentation is easier using blog posts as they represent the natural utterances produced by a blogger. Third, a post index allows for simple incremental indexing and does not require frequent re-computations of pseudo-documents that are meant to represent an entire blog.

In this chapter, we introduce three models that are able to rank blogs for a given query based on a post index. (i) The *Blogger model* is blog-based and tries to estimate the relevance of the blog based on all its posts. (ii) The *Posting model* is post-based and first ranks individual posts, after which it tries to estimate a blog's relevance from the post scores. (iii) The *two-stage model* exploits the following observation about human strategies for identifying complex information objects such as blogs (or people, for that matter). Prior to in-depth examination of complex information objects, humans display exploratory search behavior triggered by salient features of such objects [98]. This insight gives rise to the following two-stage model for blog feed search: In stage 1, we take individual utterances (i.e., posts) to play the role of "attention triggers" and select an initial sample of blogs based on the most interesting (in this case, relevant) posts given the query, using a post-based approach. Then, in stage 2, we only consider these most interesting blogs, which we then examine more in-depth by considering all their posts to determine the likelihood of the topic being a central theme of the blog, using a blog-based approach.

All models use associations between posts and blogs to indicate to which blog their relevance score should contribute. The models achieve highly competitive retrieval performance (on community-based benchmarks), although the Blogger model consistently outperforms the Posting model in terms of retrieval effectiveness while the Posting model needs to compute substantially fewer associations between posts and blogs and, hence, is more efficient. The two-stage model, subjected to additional pruning techniques, maintains (and even increases) effectiveness compared to the Blogger model, while improving on efficiency.

The research questions we address in this chapter are the following:

RQ 2 Can we effectively and efficiently search for people who show a recurring interest in a topic using an index of utterances?

1. Can we model the task of blogger finding as an association finding task?
2. How do our implementations of the post-based (Posting) and blog-based (Blogger) models compare to each other in terms of retrieval effectiveness and efficiency?
3. Can we introduce different association strength indicators between posts and blogger and how do they influence performance?
4. Can we combine the strengths of the two models and how does this new, two-stage model perform compared to our baselines?

5. Can we improve efficiency by limiting the number of posts we look at or by reducing the document representations (e.g., title-only)?

The remainder of this chapter is organized as follows. The three retrieval models that we use are discussed in Section 5.1. Our experimental setup is detailed in Section 5.2 and our baseline results are established in Section 5.3. Results on our two-stage model and its refinements are presented in Section 5.4. A discussion (Section 5.5) and conclusion (Section 5.6) complete the chapter.

5.1 Probabilistic Models for Blog Feed Search

In this section we introduce three models for blog feed search, i.e., for the following task: given a topic, identify blogs (that is, feeds) about the topic. The blogs that we are aiming to identify should not just mention the topic in passing but display a recurring central interest in the topic so that readers interested in the topic would add the feed to their feed reader.

To tackle the task of identifying such key blogs given a query, we take a probabilistic approach, similar to the language modeling approach introduced in Section 3.3. We formulate the task as follows: *what is the probability of a blog (feed) being a key source given the query topic Q?* That is, we determine $P(blog|Q)$ and rank blogs according to this probability. Since the query is likely to consist of very few terms to describe the underlying information need, a more accurate estimate can be obtained by applying Bayes' Theorem, and estimating:

$$P(blog|Q) = \frac{P(Q|blog) \cdot P(blog)}{P(Q)}, \tag{5.1}$$

where $P(blog)$ is the probability of a blog and $P(Q)$ is the probability of a query. Since $P(Q)$ is constant (for a given query), it can be ignored for the purpose of ranking. Thus, the probability of a blog being a key source given the query Q is proportional to the probability of a query given the blog $P(Q|blog)$, weighted by the *a priori* belief that a blog is a key source, $P(blog)$:

$$P(blog|Q) \propto P(Q|blog) \cdot P(blog). \tag{5.2}$$

Since we focus on a post-based approach to blog feed search, we assume the prior probability of a blog $P(blog)$ to be uniform. The search task then boils down to estimating $P(Q|blog)$, the likelihood of a blog generating query Q.

In order to estimate the probability $P(Q|blog)$, we adapt generative probabilistic language models used in information retrieval in three different ways. In our first model, the Blogger model (Section 5.1.1), we build a textual representation of a blog, based on posts that belong to the blog. From this representation we estimate the probability of the query topic given the blog's model. Our second model, the Posting model (Section 5.1.2), first retrieves individual blog posts that are relevant to the query, and then considers the blogs from which these posts originate. Finally, we introduce a two-stage approach in Section 5.1.3, in which we use the Posting model to find "attention triggers" (i.e., blog posts) from which an initial set of blogs is selected. Stage 2 then explores these blogs in-depth using the Blogger model.

The Blogger model and Posting model originate from the field of expert finding and correspond to Model 1 and Model 2 [15, 18]. We opt for translating these models to the new setting of blog feed search and focus on using blog specific associations, combining the models, and improving efficiency. In the remainder of this chapter we use the open source implementation of both the Blogger and Posting model, called EARS:[3] Entity and Association Retrieval System.

5.1.1 Blogger model

The Blogger model estimates the probability of a query given a blog by representing the blog as a multinomial probability distribution over the vocabulary of terms. Therefore, a blog model $\theta_{blogger}(blog)$ is inferred for each blog, such that the probability of a term given the blog model is $P(t|\theta_{blogger}(blog))$. The model is then used to predict how likely a blog would produce a query Q. Each query term is assumed to be sampled identically and independently. Thus, the query likelihood is obtained by taking the product across all terms in the query:

$$P(Q|\theta_{blogger}(blog)) = \prod_{t \in Q} P(t|\theta_{blogger}(blog))^{n(t,Q)}, \tag{5.3}$$

where $n(t, Q)$ denotes the number of times term t is present in query Q.

To ensure that there are no zero probabilities due to data sparseness, it is standard to employ smoothing. That is, we first obtain an empirical estimate of the probability of a term given a blog $P(t|blog)$, which is then smoothed with the background collection probabilities $P(t)$:

$$P(t|\theta_{blogger}(blog)) = (1 - \lambda_{blog}) \cdot P(t|blog) + \lambda_{blog} \cdot P(t). \tag{5.4}$$

In Equation 5.4, $P(t)$ is the probability of a term in the document repository. In this context, smoothing adds probability mass to the blog model according to how likely it is to be generated (i.e., published) by any blog.

To approximate $P(t|blog)$ we use the blog's posts as a proxy to connect the term t and the blog in the following way:

$$P(t|blog) = \sum_{post \in blog} P(t|post, blog) \cdot P(post|blog). \tag{5.5}$$

We assume that terms are conditionally independent from the blog (given a post), that is, $P(t|post, blog) = P(t|post)$. We approximate $P(t|post)$ with the standard maximum likelihood estimate, i.e., the relative frequency of the term in the post. Our first approach to setting the conditional probability $P(post|blog)$ is to allocate the probability mass uniformly across posts, i.e., assuming that all posts of the blog are equally important. In Section 5.4 we explore other ways of estimating this probability.

[3] `http://code.google.com/p/ears`

We set the smoothing parameter as follows: $\lambda_{blog} = \beta/(|blog|+\beta)$ and $(1-\lambda_{blog}) = |blog|/(|blog| + \beta)$, where $|blog|$ is the size of the blog model, i.e.:

$$|blog| = \sum_{post \in blog} |post| \cdot P(post|blog), \tag{5.6}$$

where $|post|$ denotes the length of the post. This way, the amount of smoothing is proportional to the information contained in the blog; blogs with fewer posts will rely more on the background probabilities. This method resembles Bayes smoothing with a Dirichlet prior [121]. We set β to be the average blog length in the collection; see Table 5.3 for the actual values used in our experiments.

5.1.2 Posting model

Our second model assumes a different perspective on the process of finding blog feeds. Instead of directly modeling the blog, individual posts are modeled and queried (hence the name, Posting model); after that, blogs associated with these posts are considered. Specifically, for each blog we sum up the relevance scores of individual posts, that is, $P(Q|\theta_{posting}(post)))$, weighted by their relative importance given the blog, that is, $P(post|blog)$. Formally, this can be expressed as:

$$P(Q|blog) = \sum_{post \in blog} P(Q|\theta_{posting}(post)) \cdot P(post|blog). \tag{5.7}$$

Assuming that query terms are sampled independently and identically, the probability of a query given an individual post is:

$$P(Q|\theta_{posting}(post)) = \prod_{t \in Q} P(t|\theta_{posting}(post))^{n(t,Q)}. \tag{5.8}$$

The probability of a term t given the post is estimated by inferring $P(t|\theta_{posting}(post))$, a post model, for each post following a standard language modeling approach:

$$P(t|\theta_{posting}(post)) = (1 - \lambda_{post}) \cdot P(t|post) + \lambda_{post} \cdot P(t), \tag{5.9}$$

where λ_{post} is set proportional to the length of the post, $|post|$, such that $\lambda_{post} = \beta/(|post| + \beta)$ and $(1 - \lambda_{post}) = |post|/(|post| + \beta)$. In this way, short posts receive more smoothing than long ones. We set the value of β to be equal to the average post length in the collection; again, see Table 5.3 for the actual numbers used in our experiments.

5.1.3 A two-stage model

We also consider a two-stage model that integrates the Posting model, which is the more efficient of the two, as we will see, and the Blogger model, which has a better representation of the blogger's interests, into a single model. To achieve this goal, we use two separate stages:

Stage 1: Use Equation 5.8 to retrieve blog posts that match a given query and construct a truncated list B of blogs to which these posts belong. We do not need to "store" the ranking of this stage.

Stage 2: Given the list of blogs B, we use Equation 5.3 to rank just the blogs that are present in this list.

By limiting, in stage 1, the list of blogs B, that need to be ranked in stage 2, this two-stage approach aims at improving efficiency, while it maintains the ability to construct a ranking based on the complete profile of a blogger.

More precisely, let N, M be two natural numbers. Let f be a ranking function on blog posts: given a set of posts it returns a ranking of those posts; f could be recency, length, or it could be a topic dependent function, in which case the query Q needs to be specified. We write $(f \upharpoonright N)(blog)$ for the list consisting of the first N posts ranked using f; if Q is a query, we write f_Q for the post ranking function defined by Equation 5.8. Then,

$$P(Q|\theta_{two}(blog)) = \begin{cases} 0, & \text{if } (f_Q \upharpoonright N)(blog) = \emptyset \\ \prod_{t \in Q} P(t|\theta_{two}(blog))^{n(t,Q)}, & \text{otherwise,} \end{cases} \tag{5.10}$$

where $(f_Q \upharpoonright N)(blog)$ denotes the set of top N relevant posts given the query and $\theta_{two}(blog)$ is defined as a mixture, just like Equation 5.4:

$$P(t|\theta_{two}(blog)) = (1 - \lambda_{blog}) \cdot P_{two}(t|blog) + \lambda_{blog} \cdot P(t), \tag{5.11}$$

in which the key ingredient $P_{two}(t|blog)$ is defined as a variation on Equation 5.5, restricted to the top M posts of the blog:

$$P_{two}(t|blog) = \sum_{post \in (f \upharpoonright M)(blog)} P(t|post) \cdot P(post|blog). \tag{5.12}$$

Before examining the impact of the parameters N and M in Equations 5.10 and 5.12 and more generally, before comparing the models just introduced in terms of their effectiveness and efficiency on the blog feed search task, we detail the experimental setup used to answer our research questions.

5.2 Experimental Setup

We use the test sets made available by the TREC 2007 and 2008 blog tracks for the blog feed search task. Details of this collection are discussed in Section 3.1.1 (page 26) and details on evaluation metrics used in our experiments are listed in Section 3.2 (page 28). We report on precision-oriented metrics (P5 and MRR) and mean average precision (MAP). In this section we explore the set of test topics in more detail and give detailed statistics on our indexes and smoothing parameter β.

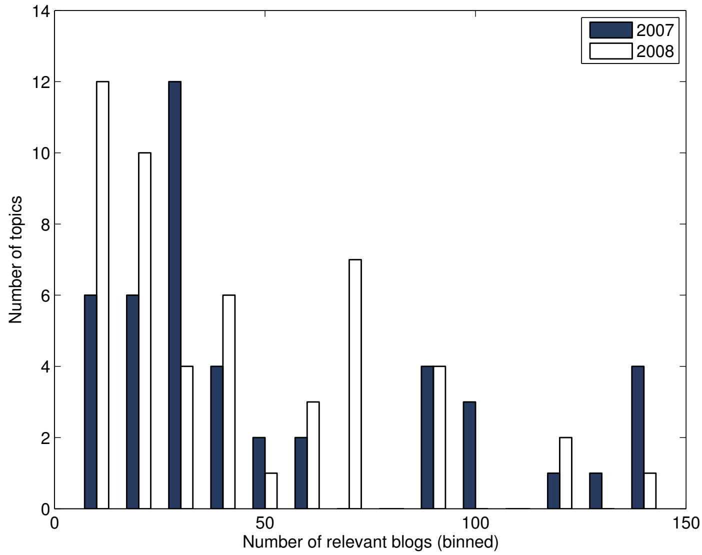

Figure 5.1: Number of relevant blogs (binned, x-axis) vs number of topics with that number of relevant blogs (y-axis).

5.2.1 Topic sets

Looking at the relevance assessments for the 2007 and 2008 TREC topics, we notice a few differences. Table 5.1 lists the statistics of the topics and relevance assessments for both years, while Figure 5.1 shows the number of topics that have a certain number of relevant blogs. To construct this plot, we made bins of 10 relevant blogs, i.e., the first point is a count of topics that have 10 or less relevant blogs in the assessments.

	2007	2008
Number of topics	45	50
Relevant results	2,221	1,943
Relevant blogs per topic (avg.)	49	39
Topics with ...		
$<$ 5 relevant blogs	0	5
$<$ 10 relevant blogs	5	11
$<$ 20 relevant blogs	12	20
$>$ 100 relevant blogs	6	3

Table 5.1: Statistics of the 2007 and 2008 topic sets.

We see that the 2008 topics have fewer relevant blogs per topic than the 2007 topics. Besides, looking at Figure 5.1 and the last 4 lines in Table 5.1, we notice that the 2008 topics are concentrated at the beginning (with a small number of relevant blogs per topic), while the 2007 topics have a later peak, and again a peak at the end of the plot ($>$

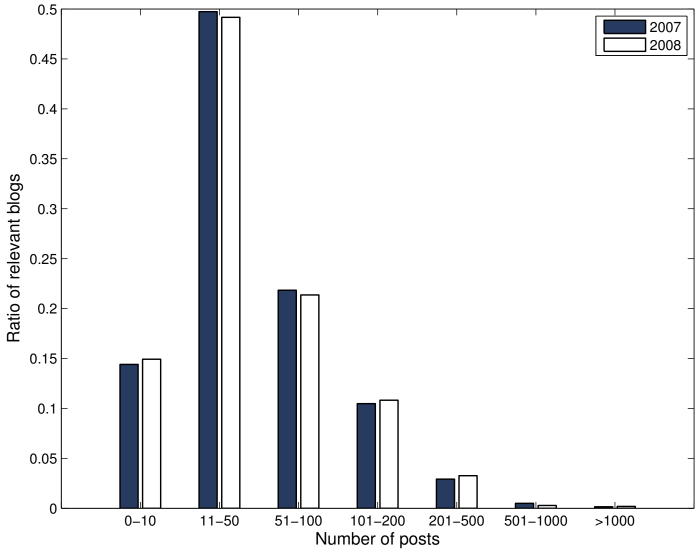

Figure 5.2: Ratio of relevant blogs (y-axis) with a certain size, in number of posts (x-axis) for both 2007 and 2008 topics.

130 relevant blogs). These differences seem to be an artifact of the topic development guidelines[4,5] used in the two years. In 2008, an additional line of instruction was added, stating that "[y]our topic area should be specific enough that there are not likely to be hundreds or thousands of relevant feeds (so 'cars' is probably too vague a topic)." This, it seems, resulted in fewer relevant blogs per topic.

We also look at the size of relevant blogs, in terms of the number of posts in a blog. In Figure 5.2 we plot how many of the relevant blogs have a certain size; unlike the number of relevant blogs, we do not observe notable differences between the two topic sets. For 2007 the average relevant blog size is 58 posts, and this is 59 posts for the 2008 topics.

5.2.2 Inverted indexes

We index the collection using the open source software package Lemur[6] (version 4.10), no stemming is applied, but we do remove stopwords. Indexing is not just done for the full (permalink) content, as described above, but we also create an index containing title-only representations of the blog posts. Here, documents are represented using just the blog post title, creating a very lean index of the collection. Index statistics are listed in Table 5.2.

[4] `http://ir.dcs.gla.ac.uk/wiki/TREC-BLOG/TREC2007`
[5] `http://ir.dcs.gla.ac.uk/wiki/TREC-BLOG/TREC2008`
[6] `http://www.lemurproject.com`

	Full content	Title-only
Number of posts	3,213,362	3,215,171
Number of blogs	83,320	83,320
Total terms	1,767,023,720	47,480,876
Unique terms	8,925,940	3,524,453
Avg. post length	550	15
Index size	13.0 GB	1.7 GB

Table 5.2: Statistics of the full content and title-only indexes.

5.2.3 Smoothing

As explained in Section 5.1, our Blogger and Posting models use smoothing, whose influence is determined using a parameter β. Since smoothing is applied at the post level for both models, we take this parameter to be the average post length (for the Blogger model, see Eq. 5.6), and we list the values of β actually used in the chapter in Table 5.3. We test the sensitivity of our models to the smoothing parameter β in Section 5.5.3.

Run		β (Blogger)	β (Posting)
All posts	Sec. 5.3.3	686	550
English posts	Sec. 5.3.3	630	506
English, no 1-post	Sec. 5.3.3	573	506
English, no 1-post, titles	Sec. 5.4.5	12	15
Comments, 50 posts	Sec. 5.4.3	595	–
Centrality, 50 posts	Sec. 5.4.3	590	–
Date, 50 posts	Sec. 5.4.3	575	–
Length, 50 posts	Sec. 5.4.3	615	–
Top 5,000 posts	Sec. 5.4.3	–	506

Table 5.3: Value of the smoothing parameter β for various runs of the Blogger and Posting model.

5.3 Baseline Results

Our aim in this section is to establish and compare our baselines, for the Blogger and Posting models. We also examine the impact of two index pruning techniques. Specifically, we look at language detection on blog posts, excluding non-English blogs, and the removal of blogs with a small number of posts and end up selecting the indexes to be used for further experiments in the chapter.

5.3.1 Language detection

The blog collection we use is a sample from the web (see Section 3.1.1) and contains not only English blogs, but also blogs written in other languages (e.g., Japanese, Chinese, and Spanish). For the task at hand we are only interested in English blogs and we would therefore like to discard all non-English blogs. To this end we apply language detection using TextCat:[7] from 3,215,171 posts we remove 640,815 posts that are labeled as non-English, leaving us with 2,574,356 posts.

5.3.2 Short blogs

The blog feed search task on which we focus requires the retrieval of blogs that have a *recurring* interest in a topic. Blogs with only one or a few posts simply cannot show a recurring interest in the topic, so ignoring them is a reasonable option and should prevent such blogs from polluting the retrieval results. In practice, we would not remove these short blogs from an index, but merely exclude blogs with fewer than K posts from our computations until they receive more posts. Potentially, this is a considerable efficiency-enhancing measure, since we do not have to care about blogs that have just started or blogs that were just "try-outs."

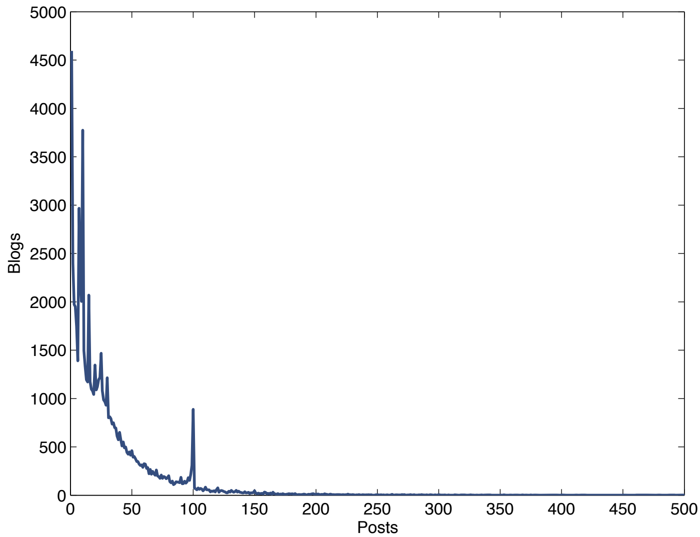

Figure 5.3: Number of posts per blog.

In Figure 5.3 we examine the distribution of the number of posts per blog in our collection, after removing non-English posts. We see that many blogs contain only a limited number of posts, with the exception for the 10, 20, 30, . . . , 100 posts. Why these peaks occur is not clear, but it is probably an artifact of the collection construction (see Section 3.1.1). A considerable number of blogs, 4,595 (∼4%), consists of a single post. We

[7] `http://odur.let.rug.nl/~vannoord/TextCat/`

do not want to exclude too many blogs, and therefore set $K = 1$, only dropping these 4,595 blogs from the index.

5.3.3 Baseline results

In Table 5.4 we list our baseline results on the blog feed search task, using the Blogger and Postings models, on the 2007 and 2008 test topics. We also consider runs that implement the additional index pruning options listed above.

Let us first consider the 2007 test topics (Table 5.4, left half). First, the Blogger and Posting models (without index pruning) perform similarly; the difference between the two runs is not significant. When we add the index pruning techniques ("English only" and "no short blogs"), we see slight improvements for the Blogger and Posting models. However, the differences are not significant when compared to the Blogger model using all posts. The best performance is achieved by the Blogger model with both index pruning techniques implemented (on MAP as well as P@5).

Turning to the 2008 test topics (Table 5.4, right half), we see that the Blogger model significantly outperforms the Posting model. Overall best performance (on all metrics) is achieved by the Blogger model with both index pruning options added.

Which posts?	**2007**			**2008**		
	MAP	P5	MRR	MAP	P5	MRR
Blogger model						
All	0.3183	0.5333	0.7159	0.2482	0.4720	0.7400
English only	0.3165	0.5333	**0.7268**	0.2469	0.4800	0.7209
English only, no short blogs	**0.3260**	**0.5422**	0.7193	**0.2521**	**0.4880**	**0.7447**
Posting model						
All	0.3104	0.5333	0.7028	0.2299$^{\triangledown}$	0.4360	0.7225
English only	0.3002	0.5067	0.6877	0.2226$^{\blacktriangledown}$	0.4160$^{\triangledown}$	0.7021
English only, no short blogs	0.3140	0.5378	0.7055	0.2305$^{\triangledown}$	0.4360	0.7237

Table 5.4: Baselines plus results of index pruning. Significance tested against Blogger model with all posts (top row).

5.3.4 Analysis

When averaged over the 2007 and 2008 topic sets, the Blogger model has just been found to be more effective than the Posting model. But averages may hide a lot of detail. Our next step, therefore, is to take a look at individual topics and compare the effectiveness of the Blogger model to the Posting model on a per-topic basis. To this end, we plot the difference in average precision between the two models, and use the scores of the Posting model as a baseline. We look at both models using the pruned index (after removal of non-English posts and short blogs). Figure 5.4 shows this plot, for the 2007 and 2008 topics.

For both years, most topics favor the Blogger model (more topics show an increase in AP over the Posting model when using the Blogger model). Table 5.5 summarizes the

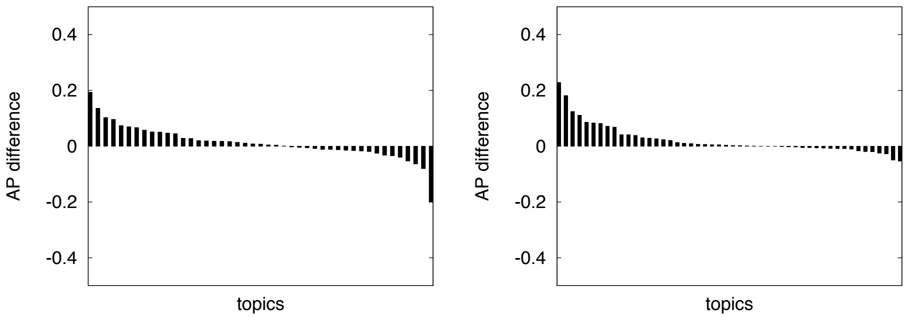

Figure 5.4: Per-topic comparison on average precision for (Left) 2007 and (Right) 2008 topics for the Posting model (baseline) and the Blogger model.

number of topics that prefer the Blogger model and the number of topics that prefer the Posting model.

Metric	**2007**		**2008**	
	Blogger	Posting	Blogger	Posting
AP	26	19	29	19
P@5	11	8	9	3
RR	9	2	8	6

Table 5.5: Number of topics that either prefer the Blogger model or the Posting model.

When explaining which topics show very different performance in AP on both models, we find the topics displayed in Table 5.6. The results in Table 5.6 suggest that on longer queries the Blogger model may be more effective than the Posting model. To explore this hypothesis in more detail, we group AP differences by query length; see Figure 5.5. We see that, on average, the Blogger model outperforms the Posting model when the query consists of at least two words. We also see that on single term queries, the Posting model slightly outperforms the Blogger model on average AP.

In order to quantify to which extent the two models—Blogger and Posting—identify different relevant blogs, we count the number of unique retrieved, relevant blogs for each model over the whole set of topics. Table 5.7 lists the number of relevant blogs retrieved by one model, that are not returned by the other model (in the top 100 results).

The results indicate that the Blogger model is better at retrieving "new" relevant blogs, but that the Posting model is also capable of retrieving unique relevant blogs. This suggests that a combination of the two models may well outperform both models individually. We explore these uniquely retrieved blogs in more detail and look at the size of the blogs (viz. Section 5.2.1), and list results in Table 5.8.

The blogs retrieved only by the Blogger model are comparable in size to the average size of relevant blogs (58 posts); the average size of blogs retrieved only by the Posting model, however, is much smaller. It seems the Blogger model becomes more useful with

Topic	Increase	Model
machine learning (982)	0.2000 (25%)	Posting
photography (983)	0.0635 (44%)	Posting
dlsr camera review (984)	0.1936 (42%)	Blogger
buffy the vampire slayer (993)	0.1358 (69%)	Blogger
organic food and farming (1082)	0.1816 (46%)	Blogger
veronica mars (1091)	0.2286 (36%)	Blogger

Table 5.6: Topics with large difference in AP between Blogger and Posting model. The column labeled "Model" indicates which model performs best. (The number in brackets is the topic ID.)

Model	2007	2008
Blogger	100	96
Posting	76	57

Table 5.7: The number of unique relevant blogs for the Blogger and Posting model in the top 100 results.

growing blog sizes, while the Posting model is stronger for smaller blogs.

Model	2007	2008
Blogger	52	56
Posting	37	43

Table 5.8: The average size (in posts) of unique relevant blogs for both models.

5.3.5 Intermediate conclusions

We can achieve good performance on the blog feed search task, using a post index and models based on association finding models originally developed for expert finding. To substantiate this claim we compare the effectiveness of our models to that achieved by TREC participants [119, 120]. For 2007, both our models would have been ranked second on MAP and around the median for MRR. On the 2008 topics, our models are ranked in the top 5 for both MAP and MRR. Since we are still only looking at baselines of our models and comparing these to considerably more advanced approaches (that use, e.g., query expansion or link structure), we conclude that our models show good effectiveness on the task of blog feed search.

Comparing the Blogger and Posting model, we see that the Blogger model performs better, with significant differences for the 2008 topics. Finally, combining the two index pruning techniques—removing non-English blogs and blogs consisting of a single post—helps to improve not just the efficiency of our models but also their effectiveness.

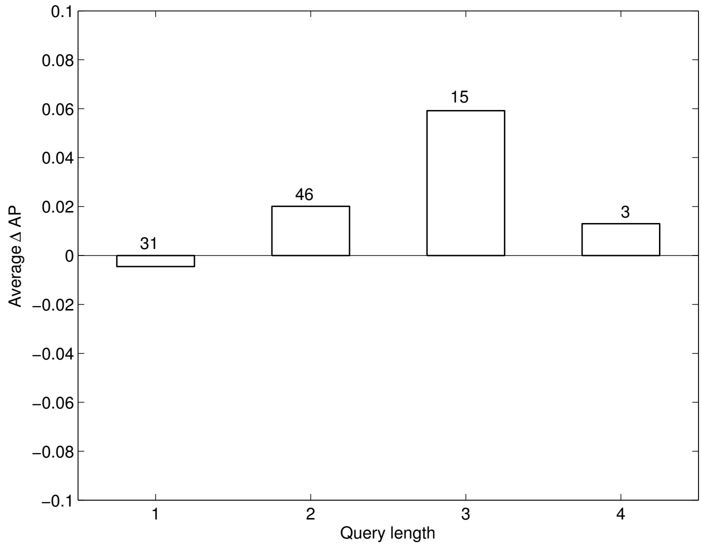

Figure 5.5: Average improvement in AP for the Blogger model over the Posting model, grouped by query length. The number above the columns indicate the number of topics of that length.

Based on these findings, we continue our experiments in the following sections using an index created from English-only posts and without short blogs. The statistics of this index are given in Table 5.9.

Index	Posts	Blogs	Avg. posts per blog
All posts	3,215,171	83,320	39
All English posts	2,574,356	76,358	34
English, no short blogs	2,569,761	71,763	36

Table 5.9: Statistics of full content indexes used in the chapter.

5.4 A Two-Stage Model for Blog Feed Search

Given the size of the blogosphere, efficiency is an important concern when addressing tasks such as blog feed search and blog post retrieval. Having introduced models that can use a single index for both tasks is a first step in achieving efficient, yet effective solutions. In Section 5.3 we took a second step and explored ways of index pruning to improve efficiency, while keeping effectiveness at a competitive level.

In this section we continue to look for ways of enhancing efficiency in our models while determining the impact of these enhancements on retrieval effectiveness. We do so by combining the strengths of the Blogger and Posting models into a two-stage model where the Posting model is used to identify a limited set of potentially valuable blog feeds for a given topic and then the Blogger model is used to construct a final ranking of this selection, as specified in Section 5.1.3. In each of the two stages we work with cut-offs on the number of posts or blogs considered.

We start by motivating the two-stage model in more detail. We then consider notions of post importance that can be used for cut-offs. Next, we consider the impact of cut-offs on the effectiveness of the single stage Blogger and Posting models before combining them. We conclude the section with a further enhancement of the two-stage model using a very lean representation of the contents of blogs and their posts.

5.4.1 Motivation

We have seen that the Blogger model is more effective at the task of blog feed search than the Posting model. One clear disadvantage of the Blogger model is that it needs to be computed by considering a large numbers of associations $P(post|blog)$ (cf. Eq. 5.5). What if we could restrict both the blogs and posts that we need to consider without negatively impacting the Blogger model's effectiveness? Our two-stage model uses the Posting model for pre-selecting blogs that are then fed to the Blogger model to produce the final ranking. To increase the efficiency of the Posting model, we restrict the number of blogs that it needs to consider (see Eq. 5.10) and to further increase the efficiency of the subsequent ranking step by the Blogger model, we restrict the number of posts to consider per blog (see Eq. 5.12).

To get an idea of the efficiency enhancement that may be obtained by using this two-stage approach, we look at the number of associations that need to be considered. Using the settings employed in our experiments below, after the Posting model stage, we are left with an average of 1,923 blogs per topic. In the second stage, the Blogger model uses *at most* 50 posts per blog. In our experiments below, this leads to a maximum of 96,150 associations that have to be considered for each test topic. Table 5.10 shows the numbers of associations that need to be looked at by the Blogger model, when it takes all posts into account, only 50 per blog, only 10 per blog, or when it functions as the second stage in the two-stage model with the settings just given. Clearly, then, substantial efficiency improvements can be gained by the two-stage model over the original Blogger model.

Setting	Associations	% of all
Blogger, all posts per blog	2,569,761	100%
Blogger, 50 posts per blog	1,839,268	72%
Blogger, 10 posts per blog	643,252	25%
Two-stage model	**96,150**	**4%**

Table 5.10: Number of associations that needs to be considered over all topics; in the two-stage model (bottom row) 1,923 blogs are pre-selected by the Posting model (per test topic, on average) and for each of these, the Blogger model considers at most 50 posts.

5.4.2 Estimating post importance

Now that we have seen that cut-offs can substantially reduce the number of associations that need to be considered when computing the models, we investigate a number of ways of ranking posts (from a single blog) with respect to their importance to their parent blog; cut-offs as implemented in using the restricted summation in Eq. 5.12 will be based on these importance rankings. Estimating post importance in blogs should ideally make use of blog specific features. In the following paragraphs we introduce three blog-specific features.

Post length. Blog posts are characterized by their relatively small size in terms of number of words. Short blurbs on what a blogger did today or what she is currently doing make up for many of the blog posts in the blogosphere. We are interested in the posts that contain more information than just these blurbs. We translate this into a preference for longer blog posts and assign higher association strengths to longer posts, viz. Eq. 5.13:

$$P(post|blog) = \frac{\log(|post|)}{\sum_{post' \in blog} \log(|post'|)} \tag{5.13}$$

where $|post|$ is the length of the post in words.

Centrality. In determining the recurring interest of a blog, we are interested in blog posts that are central to a blog. That is, we want to emphasize posts that differ least from the blog as a whole and thereby represent the "core" of a blog. We estimate the centrality using the KL-divergence between each post and the blog as a whole (Eq. 5.14).

$$KL(post||blog) = \sum_{t} P(t|post) \cdot \frac{P(t|post)}{P(t|blog)}. \tag{5.14}$$

Since a lower KL-divergence indicates a more central blog post, we take the inverse of the KL divergence as the centrality score for a post, and normalize over all posts for a given blog to arrive at the association strength of a post:

$$P(post|blog) = \frac{KL(post||blog)^{-1}}{\sum_{post' \in blog} KL(post'||blog)^{-1}}. \tag{5.15}$$

Comments. Explicitly marked up social interactions are very characteristic for the blogosphere: bloggers allow readers to comment on what they have written and sometimes get involved in the discussion. We build on the intuition that posts that receive many comments are more likely to be of interest to readers, since many readers before them took the effort of leaving behind a comment. We turn the number of comments received by a post into a reflection of its importance; see Eq. 5.16:

$$P(post|blog) = \frac{1 + \log(|comm(post)| + 1)}{\sum_{post' \in blog}(1 + \log(|comm(post')| + 1)}, \tag{5.16}$$

where $|comm(post)|$ is the number of comments received by $post$. To make sure the $\log$ is defined, we add one comment before taking the log; we add one comment again after this, to prevent zero probabilities. To estimate the number of comments per post, we build on the observation that comments on blog posts follow a similar pattern across different posts: All comments consist of an author, actual comment content, and a timestamp. We use a part of this pattern, the timestamps, and count the number of occurrences of these in a blog post. Manual assessment of several samples revealed that this is a good indicator of the actual number of comments.

Other social aspects of the blogosphere, the blogroll and permalinks, are not considered here, but could also be of interest: blogs that are mentioned a lot in blogrolls could be of more interest, while a larger number of permalinks to a post could also reflect post importance.

5.4.3 Pruning the single stage models

With multiple notions of post importance in place, we examine the impact on retrieval effectiveness of pruning the computations to the top N posts ordered by importance (according to one of the notions of importance). In this section we do not aim at obtaining the highest scores, but focus on the influence of pruning on retrieval performance for both models.

Both baseline models—Blogger and Posting—offer a natural way of improving efficiency: the Blogger model allows one to limit the number of posts to be taken into account for estimating the model; that is, instead of Equation 5.5, we compute

$$P(t|blog) = \sum_{post \in (f \restriction N)(blog)} P(t|post) \cdot P(post|blog),$$

where $(f \restriction N)(blog)$ is a restricted set of posts. In the Posting model we can similarly limit ourselves to a small number of posts when aggregating scores, using

$$P(Q|blog) = \sum_{post \in (f_Q \restriction N)(blog)} P(Q|\theta_{posting}(blog)) \cdot P(post|blog)$$

instead of Equation 5.7. Below, we explore the impact of these efficiency improvements on the retrieval effectiveness; we take the top N posts, ranked using the importance factors provided above.

Blogger model. Here, we can vary the number of posts to include when constructing the model of a blog. Besides looking at the obvious recency ordering of posts before pruning (newest to oldest post), we also look at the blog importance features considered above: comments, centrality, and post length. We order the list of posts for each blog based on each of these features and prune the list to at most N posts. Figure 5.6 shows the performance in terms of MAP for the various ways of ordering and for multiple values of N.

The plots show that we can improve effectiveness on MAP by limiting the number of posts we take into account when constructing the Blogger model, an insight that we will use in setting up the two-stage model below. Even more interesting is the fact that the "original" ordering (by recency) is outperformed by other ways of ordering posts, especially ordering by post length. Table 5.11 displays the number of associations (i.e.,

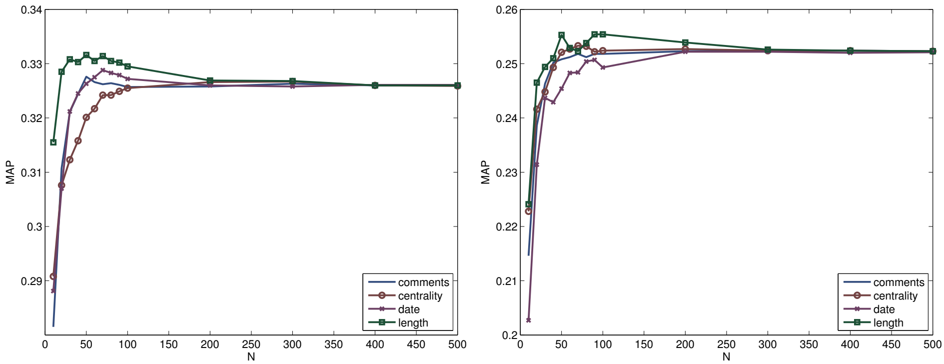

Figure 5.6: Influence of selecting at most N posts on MAP of the Blogger model for (Left) 2007 and (Right) 2008, where posts are ordered by recency, comments, centrality, or length.

$P(post|blog)$ values) that need to be considered for different values of N and shows that by pruning the post list, we substantially reduce this number.

N	Associations	% of all
all	2,569,761	100%
500	2,510,802	98%
100	2,281,165	89%
50	**1,839,268**	**72%**
20	1,095,378	43%
10	643,252	25%

Table 5.11: Number of associations that need to be considered when up to N posts are used for creating a Blogger model (regardless of ordering).

Table 5.12 shows the effectiveness of limiting the number of posts used to construct the Blogger model to 50, for various ways of ordering the posts. We observe that most orderings show no significant difference compared to the using all posts.

Posting model. Next we explore the impact of pruning on the effectiveness of the Posting model. In Figure 5.7 we plot the number of posts that are taken into account when aggregating post scores into blog scores against the various metrics for both topic sets. From the plots we observe that we do not need to take all posts into account when scoring blogs. Rather, we can do with only a relative small number of posts—again, an insight that we will use in setting up the two-stage model below.

Table 5.13 lists the effectiveness of pruning the post list for the Posting model. Even though the best performance is achieved using all posts, scores after pruning the list to 5,000 posts are promising. Given the efficiency improvement we achieve by going back from over 2.5M posts to only 5,000, we feel that this drop in effectiveness is defendable.

Ordering	2007			2008		
	MAP	P5	MRR	MAP	P5	MRR
– (all posts)	0.3260	0.5422	0.7193	0.2521	0.4880	0.7447
Recency	0.3263	**0.5600**	0.7110	0.2454$^{\triangledown}$	0.4840	0.7423
Centrality	0.3201$^{\triangledown}$	0.5333	0.7081	0.2521	0.4880	0.7632
Comments	0.3276	0.5556	**0.7422**	0.2508	**0.5000**	0.7351
Length	**0.3316**	0.5467	0.7310	**0.2553**	0.4960	**0.7665**

Table 5.12: Results on the blog feed search task of the Blogger model built using at most top 50 posts, under various orderings. Significance tested against all posts (top row).

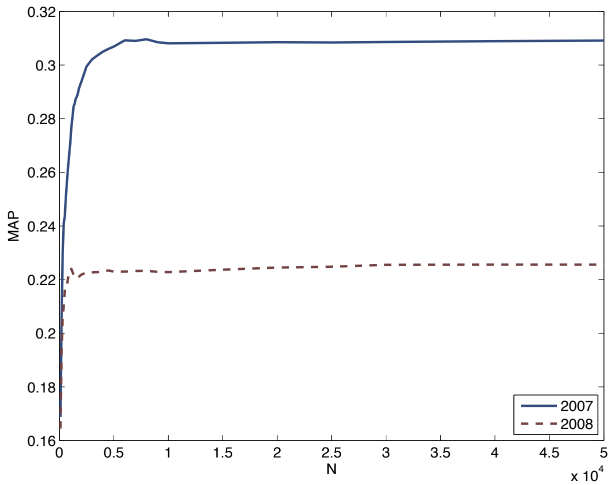

Figure 5.7: Impact of limiting to the top N posts on MAP of the Posting model.

As an aside, we explored using the three blog characteristics (comments, centrality, and post length) as estimates of the association strength in the Posting model, and its influence on pruning. Results, however, did not show an improvement over a uniform probability.

The values of 50 (for the Blogger model) and 5,000 (for the Posting model) were obtained by using one year as the training set and the other as the test set and averaging the optimal outcomes.

5.4.4 Evaluating the two-stage model

We quickly turn to the results achieved by the two-stage model as defined in Section 5.1.3. Table 5.14 lists the results of four settings, three of which we have already discussed: (i) the Blogger model (all posts), (ii) the Blogger model with 50 posts (length ordered), (iii) the Posting model with $5,000$ posts, and (iv) the two-stage model using items (ii), and (iii) as components (that is, with $N = 5,000$ and $M = 50$ in Eq. 5.10 and 5.12, respectively).

N	2007			2008		
	MAP	P5	MRR	MAP	P5	MRR
2,569,761 (all)	**0.3140**	**0.5378**	**0.7055**	**0.2305**	0.4360	0.7237
10,000	0.3081▼	0.5244	0.6907	0.2228▼	0.4360	0.7229
5,000	0.3069▽	0.5289	0.6912	0.2230▼	0.4320	0.7232
1,000	0.2712▼	0.5156	0.6821	0.2232	**0.4440**	**0.7403**
100	0.1688▼	0.4489▼	0.6729	0.1645▼	0.4120	0.6980

Table 5.13: Results on the blog feed search task of the Posting model, with pruning, selecting only the top N posts. Significance tested against the all posts runs (top row).

Setting	2007			2008		
	MAP	P5	MRR	MAP	P5	MRR
Blogger (all)	0.3260	0.5422	0.7193	0.2521	0.4880	0.7447
Blogger (top 50)	0.3316	**0.5467**	0.7310	0.2553	0.4960	**0.7665**
Posting (top 5,000)	0.3069▽	0.5289	0.6912	0.2230▼	0.4320	0.7232
Two-stage model	**0.3334**	**0.5467**	**0.7321**	**0.2566**	**0.5040**	**0.7665**

Table 5.14: Results on the blog feed search task of the combined approach. Significance tested against the baseline (i.e., top row).

The results show that our two-stage model is not significantly different than the Blogger model, but it does lead to an increase in effectiveness.

5.4.5 A further reduction

In Section 5.2.2 we introduced two document representations of the blog posts in our collection: A full content representation, *full*, and a title-only representation, *title*. The title-only representation is much smaller in terms of disk space and average document length, and is therefore more efficient to search in than the full content representation. In this section we explore the effects of using various (combinations of) document representations in our two-stage model.

We compare four combinations of the two representations: (i) full content for both stages, (ii) title-only for the Posting model (stage 1), full content for the Blogger model (stage 2), (iii) full content for the Posting model (stage 1), title-only for the Blogger model (stage 2), and (iv) title-only in both stages. The results of these combinations are displayed in Table 5.15.

For the 2007 topics the run using a title-only representation in stage 1, and the full content in stage 2 performs best on P5 and MRR; the 2008 topics show a slightly mixed result, with no clear difference between full content representations in both stages and title-only in stage 1 and full content in stage 2. What do these results mean? Using a lean title-only document representation in stage 1, the Posting model, seems sufficient to select the right blogs. In stage 2 however, we need a full content representation to construct blog models and use these to rank the blogs.

Stage 1 (Posting)	Stage 2 (Blogger)	2007 MAP	2007 P5	2007 MRR	2008 MAP	2008 P5	2008 MRR
full	full	0.3334	0.5467	0.7321	**0.2566**	**0.5040**	0.7665
title	full	**0.3556**	**0.6533**▲	**0.8574**▲	0.2415	0.4840	**0.7794**
full	title	0.2719▼	0.6178	0.7816	0.1995▼	0.4776	0.7125
title	title	0.2601▼	0.6133	0.7810	0.1889▼	0.4640	0.6983

Table 5.15: Results on the blog feed search task of different document representations in the two-stage model. Significance tested against the best performing settings using full content for both stages (top row).

5.4.6 Per-topic analysis of the two-stage model

To better understand the performance of the two-stage model, we compare the runs using different document representations to a baseline, the Blogger model. We plot the baseline as the "zero" line, and plot for each topic the difference in average precision for two ways of combining the models, full+full and title+full (see Table 5.15 for the average results). The plots are given in Figure 5.8.

We can see that for the full+full document representation, improvements are modest, with slightly more topics improving over the baseline than not. The results for the title+full run are more outspoken: we see a lot of 2007 topics with a steady improvement over the baseline, whereas for the 2008 topics there appears to be a tendency towards a decrease in performance compared to the Blogger model. We provide a different perspective on the matter by listing the number of topics that shows either an increase or decrease in performance over the Blogger model baseline; see Table 5.16. We see that the combined title+full model increases performance in terms of AP for most 2007 topics, while hurting only a few of them. In terms of reciprocal rank, the title+full run has equal performance to the Blogger baseline for most topics, but also achieves an increase for 15 topics. As to 2008, more topics are hurt than helped according to AP, while the balance is positive for P5 and RR.

Run	2007 AP ⇑	2007 AP ⇓	2007 P5 ⇑	2007 P5 ⇓	2007 RR ⇑	2007 RR ⇓	2008 AP ⇑	2008 AP ⇓	2008 P5 ⇑	2008 P5 ⇓	2008 RR ⇑	2008 RR ⇓
full+full	27	17	4	3	3	3	29	21	6	2	5	4
title+full	32	13	23	5	15	1	23	25	13	9	13	3
title+title	13	32	21	12	12	11	15	34	14	13	9	14

Table 5.16: Number of topics where performance goes "up" (⇑) or "down" (⇓) compared to the Blogger baseline.

Next, we take a closer look at which topics improve most on any of the metrics with respect to the baseline, when we use the two-stage model with the title-only representation in the first stage. Table 5.17 shows these topics. It is interesting to examine the number of relevant retrieved blogs per topic for the Blogger model and for the two-stage

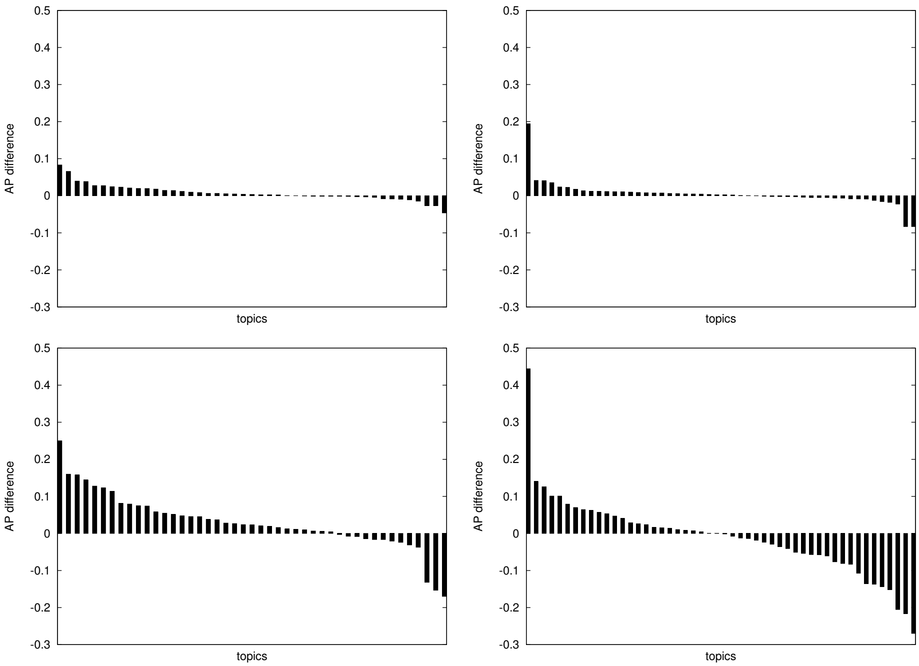

Figure 5.8: Per-topic comparison for (Left) 2007 and (Right) 2008 topics on average precision (AP) for the baseline (Blogger model) compared to the two-stage model using (Top) full+full and (Bottom) title+full. Positive bars indicate better performance by the two-stage model, negative bars indicate better performance by the Blogger model.

model. From the top improving topics, displayed in Table 5.17, only topics 968 and 988 have more relevant results retrieved by the two-stage model. The other topics get their improvements from an improved ranking. Topic 993 (*buffy the vampire slayer*) loses 11 relevant blogs in the two-stage model (reflected in a drop in AP), but still improves a lot on the precision metrics. Over all topics, the Blogger model finds 179 more relevant blogs than the two-stage model (9%), but the two-stage model is, in general, better at ranking the relevant blogs higher. This is reflected in Figure 5.9, where we see that (especially for 2008) the Blogger model retrieves more relevant blogs for most topics than the two-stage model.

The differences in the number of retrieved relevant blogs are also reflected in the number of unique relevant blogs for the Blogger model and the two-stage model. Table 5.18 shows that both models are capable of retrieving relevant blogs that are ignored by the other model. Interestingly, the unique blogs retrieved by the two-stage model are contain much posts than the unique results of the Blogger model.

Finally, we look at the influence of the two-stage model on queries of different length, as we did in Figure 5.5. In this case, we compare results between the baseline Blogger model, and the two-stage model, and group the difference in AP by query length. The results in Figure 5.10 show that the two-stage model outperforms the Blogger model on

Topic	ΔAP	ΔP5	ΔRR
christmas (968)	0.0378	0.4000	0.6667
robot companions (988)	0.1599	0.4000	0.2500
lost tv (990)	0.2496	0.2000	0.5000
buffy the vampire slayer (993)	-0.0311	0.6000	0.8333
celebrity babies (1078)	0.4444	0.2000	0.8889
3d cities globes (1086)	0.0164	0.2000	0.6667

Table 5.17: Topics that show an increase in performance on any metric going from the baseline to the two-stage model (title+full). (The number in brackets is the topic ID.)

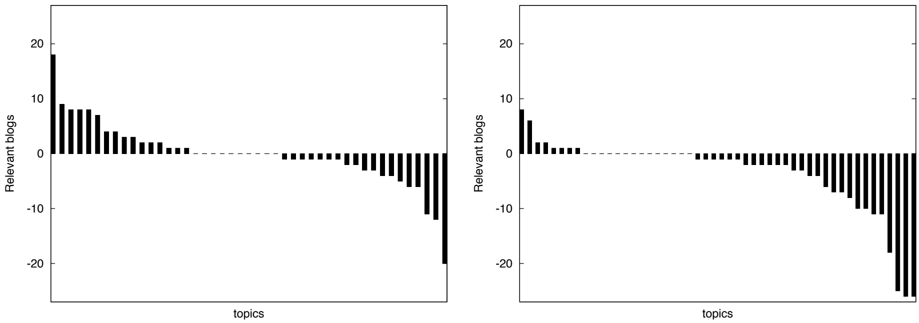

Figure 5.9: Per-topic comparison for (Left) 2007 and (Right) 2008 topics on the number of relevant retrieved blogs for the baseline (Blogger model) and the combined model (title+full). Positive bars indicate more relevant results are retrieved by the two-stage model, negative bars indicate more relevant results are retrieved by the Blogger model.

one and two term queries, but shows a (very) slight decrease for longer queries.

5.4.7 Intermediate conclusions

The aim in this section was to examine our two-stage model, whose motivation lies in combining the Blogger model's effectiveness with the Posting model's potential for efficiency. We improved the efficiency of our models by limiting the number of posts we take into account when ranking blogs. Here, we saw that pruning post lists in the Blogger and Posting models improves efficiency, while increasing effectiveness for the Blogger model, and showing only a slight drop in effectiveness for the Posting model.

Results on our two-stage model showed that effectiveness increases when using a two-stage approach while the number of associations that need to be considered drops to just 4% of the original number of associations.

The use of a lean title-only document representation of a blog post leads to a significant drop in average post length and thus to an improvement in efficiency. Results show that using a title-only representation in stage 1 of our two-stage model (i.e., for the Posting model) is sufficient for collecting the blogs for which we need to construct

	2007		2008	
Model	uniq. blogs	size	uniq. blogs	size
Blogger (baseline)	213	31	311	39
Two-stage (title+full)	209	78	136	86

Table 5.18: The average size (in posts) of unique relevant blogs for both models.

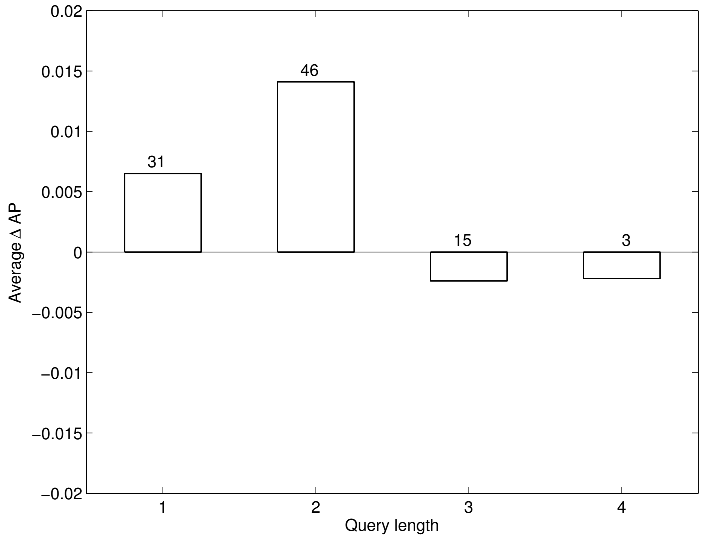

Figure 5.10: Average improvement in AP for the two-stage model (title+full) over the Blogger model, grouped by query length. The number above the columns indicate the number of topics of that length.

a blog model in stage 2 (i.e., run the Blogger model). Both efficiency and effectiveness show improvements using the two document representations in different stages of the two-stage model.

Our detailed analysis shows that by using the two-stage model we can correct for the decrease in performance of the Blogger model in comparison with the Posting model on short queries (Figure 5.5); the two-stage model improves over the Blogger model for short queries and only loses marginally on longer queries, suggesting that the two-stage model "takes the best of both worlds."

5.5 Analysis and Discussion

We reflect on the issue of efficiency vs. effectiveness of the models that we have examined, briefly touch on very high early precision functionality, and explore the impact of smoothing.

Model	posts	2007 MAP	2007 P5	2007 MRR	2008 MAP	2008 P5	2008 MRR
Blogger model							
Baseline	963,995	0.3260	0.5422	0.7193	0.2521	0.4880	0.7447
N=50/blog	598,530	0.3316	0.5467	0.7310	0.2553	0.4960	0.7665
Posting model							
Baseline	90,037	0.3140	0.5378	0.7055	0.2305	0.4360	0.7237
N=5,000/query	90,037	0.3069	0.5289	0.6912	0.2230	0.4320	0.7232
Two-stage model							
full+full	164,002	0.3334	0.5467	0.7321	0.2566	0.5040	0.7665
title+full	181,004	0.3556	0.6533	0.8574	0.2415	0.4840	0.7794

Table 5.19: Efficiency vs. effectiveness for the Blogger model, Posting model, and the two-stage model.

5.5.1 Efficiency vs. effectiveness

In this section we take a closer look at efficiency in comparison to effectiveness on the blog feed search task. Measures for effectiveness were introduced in Section 3.2.1. For measuring efficiency of our models, we look at the number of blog posts a model needs to take into account when constructing the final ranking of blogs for a given topic. In Table 5.19 we report on efficiency and effectiveness of our models.

From the results we see that pruning for the Posting model does not influence the efficiency in terms of the number of posts that are scored, since we apply pruning only after scoring posts. Here, the increase in efficiency is obtained when aggregating scores over posts: before pruning we aggregate over all 90,037 posts, after pruning we aggregate over 5,000 posts. Pruning the Blogger model shows a definite increase in efficiency, scoring 38% fewer posts after pruning. The efficiency-enhancing effects of pruning on both models directly influences efficiency of the two-stage model.

Looking at the two-stage model, we observe that the number of posts scored is 73% lower than for the Blogger model. This increase in efficiency is by no means accompanied by a decrease in effectiveness: the two-stage model maintains the Blogger model's effectiveness and even improves it.

5.5.2 Very high early precision

The well-known "I'm feeling lucky ..." search variant boils down to returning a relevant result at the top of the ranking. Our runs in Section 5.4 show (very) high early precision scores, as witnessed by the mean reciprocal rank scores. How often do they actually return a relevant result at rank 1, and if the first relevant result does not occur at rank 1, where does it occur? We look at the position of the first relevant result per topic for the 2007 and 2008 topic sets; the results are listed in Table 5.20. For most topics (80% for 2007, 67% for 2008), we do find a relevant result at rank 1. Overall, for only a small number of topics (10), we are not able to return a first relevant result in the top 4.

First relevant result	Number of topics 2007	2008
Position 1	36	34
Position 2	2	7
Position 3	3	2
Position 4	1	0
Position 5–100	3	7

Table 5.20: Number of topics grouped by the rank of the first relevant result.

Topics that prove to be particularly hard are topic 969 (*planet*), topic 991 (*U.S. Election 2008*), topic 1068 (*theater*), topic 1077 (*road cycling*), and topic 1092 (*mac os leopard*). We identify three main reasons why these topics fail to produce a relevant result in the top 4, and propose possible solutions that can be used on top of our models. In some cases the keyword descriptions of the topic are simply not specific enough for our models to be able to distinguish relevant from non-relevant blogs. This holds true for *planet*, *theater*, and *U.S. Elections 2008* (which boils down to "Elections" after query preprocessing). A possible solution to this problem is to use authoritative external sources for query expansion, as explored in Chapter 7 (adding related terms to the original query, to create a better representation of the user information need).

A second source of errors appears to be a slight mismatch between the query and the narrative that comes with it. The narrative sometimes imposes a very specific reading of the query that is not apparent from the (keyword) query itself. This is the case for *road cycling*, where many returned results talk about road cycling, but are non-relevant according to the narrative: female road cycling, personal cycling diaries, etc. One solution here would be to add terms from the description that comes with the topic to specify the topic better.

A final source of error are assessment inconsistencies. For some topics (e.g., *mac os leopard*) assessments are inconsistent: certain blogs that discuss mainly Mac OS-related topics are considered relevant (without a specific focus on the "Leopard" version of the operating system), while other blogs that do talk about the Mac OS are judged non-relevant. There is no obvious solution to this problem: it simply reflects the nature of human judgments.

5.5.3 Smoothing parameter

In Section 5.2.3 we briefly discussed the setting of the smoothing parameter β for both models. It is well known that this parameter can have a significant impact on the effectiveness of language modeling-based retrieval methods [213]. To give an impression of this impact we run a baseline experiment for our two models (comparable to the "All" runs in Section 5.3.3). We compare the automatic setting of β (as detailed in Table 5.3) to a range of different β values (1, 10, 100, 1,000, 2,000, and 5,000) and list the results in Table 5.21.

We observe that in some cases the Blogger model favors β values slightly smaller than ours. As to the Posting model, we find that our automatic setting delivers the highest

β	**2007** MAP	P5	MRR	**2008** MAP	P5	MRR
Blogger model						
1	0.3038	0.4756	0.5955	0.2303	0.4320	0.7634
10	0.3124	0.4844	0.6374	0.2400	0.4400	0.7665
100	**0.3385**	**0.5378**	0.6850	**0.2585**	0.4600	**0.7823**
686	*0.3183*	*0.5333*	***0.7159***	*0.2482*	***0.4720***	*0.7400*
1,000	0.3086	0.5289	0.7068	0,2414	0.4560	0.7069
2,000	0.2830	0.4978	0.6916	0.2256	0.4320	0.7045
5,000	0.2477	0.4489	0.6390	0.2045	0.4080	0.6590
Posting model						
1	0.2752	0.4400	0.5590	0.1983	0.4000	**0.7552**
10	0.2797	0.4844	0.5574	0.2035	0.4080	0.7491
100	0.3021	0.5200	0.6494	0.2185	0.4160	0.7360
550	***0.3104***	***0.5333***	***0.7028***	*0.2299*	*0.4360*	*0.7225*
1,000	0.3029	0.5244	0.7017	**0.2308**	0.4480	0.7014
2,000	0.2873	0.5022	0.6810	0.2239	**0.4640**	0.6731
5,000	0.2628	0.4756	0.6379	0.2069	0.4480	0.6665

Table 5.21: Impact of smoothing parameter β on effectiveness for the Blogger and the Posting model. Values corresponding to the automatic setting are typeset in italic.

scores on the 2007 topic set for all retrieval metrics. On the 2008 set, a mixed picture emerges: best MAP and P5 scores are achieved with slightly larger β values, while MRR tops when $\beta = 1$ is used. In sum, we conclude that our method of estimating the value of β based on average representation length delivers good performance across the board.

5.6 Summary and Conclusions

In this chapter we addressed the problem of supporting blog feed search and blog post retrieval from a single post-based index. In particular, we examined the balance between effectiveness and efficiency when using a post-based index for blog feed search. A Blogger and Posting model were adapted from the area of expert finding and complemented with a third, two-stage model that integrates the two. Extensive analysis of the performance of our models helps in answering the following questions:

RQ 2 Can we effectively and efficiently search for people who show a recurring interest in a topic using an index of utterances?

Our two-stage blog feed search model, complemented with aggressive pruning techniques and lean document representations, was found to be very competitive both in terms of standard retrieval metrics and in terms of the number of core operations required. As to the other two models, both the Blogger and Posting model show good performance, with the Blogger model achieving higher effectiveness and the Posting model being more efficient.

1. Can we model the task of blogger finding as an association finding task?
   We have introduced two models for blog feed search, adopted from the expert finding field, that successfully use associations between posts and blogs to construct a final ranking of blogs.

2. How do our implementations of the post-based (Posting) and blog-based (Blogger) models compare to each other on retrieval effectiveness and efficiency?
   The Blogger model consistently outperforms the Posting model on effectiveness, but the Posting model is much more efficient. Both models achieve high performance compared to other systems on a community-based benchmark.

3. Can we introduce different association strength indicators between posts and blog and how do they influence performance?
   We have explored various ways of estimating the association strength between posts and blogs. Recency appears to decrease performance over a uniform baseline, whereas the length of the post as association strength is most beneficial in terms of effectiveness.

4. Can we combine the strengths of the two models and how does this new, two-stage model perform compared to our baselines?
   We have introduced the two-stage model that is aimed at combining the Blogger and Posting model's strengths and that selects an initial set of blogs based on their relevant posts and then ranks these blogs based on (a sample of) their posts. The two-stage model consistently outperforms the Blogger model, both on effectiveness and efficiency.

5. Can we improve efficiency by limiting the number of posts we look at or by reducing the document representations (e.g., title-only)?
   For all three models, efficiency can be improved by limiting the number of posts we take into account when ranking blogs, without hurting effectiveness. Introducing a lean document representation results in further efficiency improvements and also leads to an increase of effectiveness, especially on (very) early precision.

In this chapter we have shown that we can successfully identify bloggers who demonstrate a recurring interest in a given topic using their utterances. Our two-stage model that mimics search strategies for complex objects, first locates candidate blogs by their individual posts (salient features) and then ranks these blogs by an in-depth analysis of all the blog's posts. The results of this chapter suggest that other retrieval frameworks could perform well on this task: combining various document representations, various ordering criteria, and Posting and Blogger scores seems like a typical learning to rank task (see e.g., [58, 118]). In the next chapter we move away from entering social media from the people point and focus on utterances.