UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Blog feed search with a post index

Weerkamp, W.; Balog, K.; de Rijke, M.

DOI
10.1007/s10791-011-9165-9
Publication date
2011
Document Version
Final published version
Published in
Information Retrieval

Link to publication

Citation for published version (APA):
Weerkamp, W., Balog, K., & de Rijke, M. (2011). Blog feed search with a post index. *Information Retrieval*, *14*(5), 515-545. https://doi.org/10.1007/s10791-011-9165-9

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:07 Oct 2026

Inf Retrieval (2011) 14:515–545
DOI 10.1007/s10791-011-9165-9

Blog feed search with a post index

Wouter Weerkamp · Krisztian Balog · Maarten de Rijke

Received: 18 February 2010 / Accepted: 18 February 2011 / Published online: 15 March 2011
© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract User generated content forms an important domain for mining knowledge. In this paper, we address the task of blog feed search: to find blogs that are principally devoted to a given topic, as opposed to blogs that merely happen to mention the topic in passing. The large number of blogs makes the blogosphere a challenging domain, both in terms of effectiveness and of storage and retrieval efficiency. We examine the effectiveness of an approach to blog feed search that is based on individual posts as indexing units (instead of full blogs). Working in the setting of a probabilistic language modeling approach to information retrieval, we model the blog feed search task by aggregating over a blogger's posts to collect evidence of relevance to the topic and persistence of interest in the topic. This approach achieves state-of-the-art performance in terms of effectiveness. We then introduce a two-stage model where a pre-selection of candidate blogs is followed by a ranking step. The model integrates aggressive pruning techniques as well as very lean representations of the contents of blog posts, resulting in substantial gains in efficiency while maintaining effectiveness at a very competitive level.

Keywords Blog feed search · Post-level indexing · Efficiency · Generative language models · Associations

W. Weerkamp (✉) · M. de Rijke
ISLA, University of Amsterdam, Amsterdam, The Netherlands
e-mail: w.weerkamp@uva.nl

M. de Rijke
e-mail: derijke@uva.nl

K. Balog
Department of Computer and Information Science, NTNU, Trondheim, Norway
e-mail: krisztian.balog@idi.ntnu.no

1 Introduction

We increasingly live our lives online: we keep in touch with friends on Facebook,[1] expand our network using LinkedIn,[2] quickly post messages on Twitter,[3] comment on news events on online news paper sites, help others on forums, mailing lists, or community question-answer sites, and report on experiences or give our opinions in blogs. All of these activities involve the creation of content by the end users of these platforms, as opposed to editors or webmasters. This content, i.e., user generated content, is particularly valuable as it offers an insight in what people do, think, need to know, or care about. Organizations look for ways of mining the information that is available in these user generated sources, and to do so, tools and techniques need to be developed that are capable of handling this type of content.

In this paper we focus on *blogs*. A blog is the unedited, unregulated voice of an individual (Mishne 2007), as published on a web page containing time-stamped entries (*blog posts*) in reverse chronological order (i.e., last entry displayed first). In most cases, *bloggers* (the authors of blog entries) offer readers the possibility to reply to entries in the blog (*commenting*), bloggers link to other blogs (*blogroll*), thereby creating a network of blogs, and many blogs are updated regularly. Blogs offer a unique insight in people's minds: whether the blog is about their personal life (which products do people use? what are their needs or wishes?), personal interests (what are their opinions on X?), or a more professional view on topics (can they explain X to me?), getting access to this information is valuable for many others.

Accessing the *blogosphere* (the collection of all blogs) can be done in various ways, but usually revolves around one of two main tasks: (1) identifying relevant blog posts (*blog post retrieval*), and (2) identifying relevant blogs (*blog feed search*). In (1) the goal is to list single blog posts ("utterances") that talk about a given topic; having constructed this list, one can present it to a user or use it in further downstream processing (e.g., sentiment analysis, opinion extraction, mood detection). In (2) the goal is not to return single posts, but to identify blogs that show a *recurring* interest in a given topic. Blogs that only mention the topic sporadically or in passing are considered non-relevant, but a blog (or: the person behind the blog) that talks about this topic regularly would be relevant. Again, 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, detect topic shifts in blogs). In this paper we specifically look at the second task, identifying relevant blogs given a topic, also known as blog feed search.

The total number of blogs in the world is not known exactly. Technorati,[4] 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. Most of these blogs are written in English, but the largest part of the internet users is not English-speaking. The China Internet Network Information Center (CNNIC)[5] 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

[1] http://www.facebook.com.

[2] http://www.linkedin.com.

[3] http://www.twitter.com.

[4] http://technorati.com/blogging/feature/state-of-the-blogosphere-2008/.

[5] http://www.cnnic.cn.

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, again, 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-document and indexing these pseudo-documents. In this paper, 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.

We introduce two models, the Blogger model and the Posting model, that are able to rank blogs for a given query based on a post index. Both models use associations between posts and blogs to indicate to which blog their relevance score should contribute. Both 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. To improve the efficiency of the Blogger model we integrate our Blogger and Posting models in a single two-stage model which we subject to additional pruning techniques while we maintain (and even increase) effectiveness at a competitive level.

1.1 Research questions and contributions

Our main research question is whether we can *effectively and efficiently* use a blog post index for the task of blog feed search. The Blogger and Posting models that we introduce are tested on effectiveness using standard IR methodologies. To examine their efficiency, we identify core operations that need to be executed to perform blog feed search using either of those two models.

A second set of research questions is centered around a two-stage model that we introduce to combine the strengths of the Blogger and Posting models. Specifically, we introduce a number of pruning techniques aimed at improving efficiency while maintaining (or even improving) effectiveness. We study the impact of these techniques on retrieval effectiveness as well as the impact of integrating alternative blog post representations (title-only vs. full content) into our two-stage model.

Our main contribution is twofold. First, we show that blog feed search can be supported using a post-based index. Second, we propose an effective two-stage blog feed search model together with several techniques aimed at improving its efficiency.

The remainder of this paper is organized as follows. In Sect. 2 we discuss related work on blog feed search and language modeling. The retrieval models that we use in the paper are discussed in Sect. 3. Our experimental setup is detailed in Sect. 4 and our baseline

results are established in Sect. 5. Results on our two-stage model and its refinements are presented in Sect. 6. A discussion (Sect. 7) and conclusion (Sect. 8) complete the paper.

2 Related work

In this paper related work comes in three flavors. We introduce previous research in information access in the blogosphere, we take a look at what has been done more specifically on blog feed search, and we briefly introduce language modeling for information retrieval, as this is the approach underlying our models.

2.1 Information access in the blogosphere

With the growth of the blogosphere comes the need to provide effective access to the knowledge and experience contained in the many tens of millions of blogs out there. Information needs in the blogosphere come in many flavors, addressing many aspects of blogs and thereby extending the notion of relevance, from "being about the same topic" to, for instance, "expressesing opinions about the topic" or "sharing an experience around the topic." In (Mishne and de Rijke 2006), both *ad hoc* and *filtering* queries are considered in the context of a blog search engine; the authors argue that blog searches have different intents than typical web searches, suggesting that the primary targets of blog searchers are tracking references to named entities, identifying posts that express a view on a certain concept and searching blogs that show evidence of a long-term interest in a concept.

In 2006, a blog track (Ounis et al. 2007) was launched by TREC, the Text REtrieval Conference, aimed at evaluating information access tasks in the context of the blogosphere. The first edition of the track focused mainly on finding relevant blog *posts*, i.e., on blog post retrieval, with a special interest in their opinionatedness. The 2007 and 2008 editions of the track featured a *blog distillation* or *blog feed search* task. It addresses a search scenario where the user aims to find a blog to follow or read in their RSS reader. This blog should be principally devoted to a given topic over a significant part of the timespan of the feed. Unlike blog post search tasks, the blog feed search task aims to rank blogs (i.e., aggregates of blog posts by the same blogger) instead of permalink documents.

2.2 Blog feed search

Some commercial blog search facilities provide an integrated blog search tool to allow users to easily find new blogs of interest. In (Fujimura et al. 2006), a multi-faceted blog search engine was proposed that allows users to search for blogs and posts. One of the options was to use a blogger filter: the search results (blog posts) are clustered by blog and the user is presented with a list of blogs that contain one or more relevant posts. Ranking of the blogs is done based on the EigenRumor algorithm (Fujimura et al. 2005); in contrast to the methods that we consider below, this algorithm is query-independent.

An important theme to emerge from the work on systems participating in the TREC 2007 blog feed search tasks is the indexing unit used (Macdonald et al. 2008). While the unit of retrieval is fixed for blog feed search—systems have to return blogs in response to a query—it is up to the individual systems to decide whether to produce a ranking based on a blog index or on a post index. The former views blogs as a single document, disregarding the fact that a blog is constructed from multiple posts. The latter takes samples of posts from blogs and combines the relevance scores of these posts into a single blog score. The

most effective approaches to feed distillation at TREC 2007 were based on using the (aggregated) text of entire blogs as indexing units. E.g., Elsas et al. (2008a, b) experiment with a "large document model" in which entire blogs are the indexing units and a"small document model" in which evidence of relevance of a blog is harvested from individual blog posts. They also experiment with combining the two models, obtaining best performance in terms of MAP (Arguello et al. 2008).

Participants in TREC 2007 and 2008 (Macdonald et al. 2009) explored various techniques for improving effectiveness on the blog feed search task: Query expansion using Wikipedia (Elsas et al. 2008), topic maps (Lee et al. 2008), and a particularly interesting approach—one that tries to capture the recurrence patterns of a blog—using the notion of time and relevance (Seki et al. 2007). Although some of the techniques used proved to be useful in both years (e.g., query expansion), most approaches did not lead to significant improvements over a baseline, or even led to a decrease in performance.

In the setting of blog feed search, authors have considered various ways of improving effectiveness: (1) index pruning techniques, (2) modeling topical noise in blogs to measure recurring interest, (3) using blog characteristics such as the number of comments, post length, or the posting time, and (4) mixing different document representations. We briefly sample from publications on each of these four themes.

Starting with index pruning, a pre-processing step in (Seo and Croft 2008b) consists of removing all blogs that consist of only one post, since retrieving these blogs would come down to retrieving posts and would ignore the requirement of retrieving blogs with a recurring interest. We use various types of index pruning in Sects. 5 and 6, including removing non-English blogs and blogs that consist of a single post.

As to capturing the central interest of a blog, several authors attempt to capture the central interest of a blogger by exploiting information about topical patterns in blogs. The voting-model-based approach of Macdonald and Ounis (2008) is competitive with the TREC 2007 blog feed search results reported in (Macdonald et al. 2008) and formulates three possible topical patterns along with models that encode each into the blog retrieval model. In (He et al. 2009) the need to target individual topical patterns and to tune multiple topical-pattern-based scores is eliminated; their proposed use of a coherence score to encode the topical structure of blogs allows them to simultaneously capture the topical focus at the blog level and the tightness of the relatedness of sub-topics within the blog. A different approach is proposed in (Seo and Croft 2008a), where the authors use diversity penalties: blogs with a diverse set of posts receive a penalty. This penalty is integrated in various resource selection models, where a blog is seen as a resource (collection of posts), and given a query, the goal is to determine the best resource. Below, we capture the central interest of a blogger using the KL-divergence between a post and the blog to which it belongs.

The usage of blog-specific features like comments and recency has been shown to be beneficial in blog post retrieval (Mishne 2007, Weerkamp and de Rijke 2008). In blog feed search these features can be applied in the post retrieval stage of the Posting model, but they can also be used to estimate the importance of a post for its parent blog (Weerkamp et al. 2008); we use some of these features in Sects. 5 and 6 below.

Finally, blog posts can be represented in different ways. On several occasions people have experimented with using syndicated content (i.e., RSS or ATOM feeds) instead of permalinks (HTML content) (Elsas et al. 2008a, b, Mishne 2007); results of which representation works better are mixed. Other ways of representing documents are, for example, a title-only representation, or an (incoming) anchor text representation; combinations of various representations show increased effectiveness in other web retrieval tasks

(e.g., ad hoc retrieval (Eiron and McCurley 2003, Jin et al. 2002)). We increase the efficiency of our most effective model by considering multiple content representations in Sect. 6.

2.3 Language modeling for information retrieval

At the TREC 2007 and 2008 blog tracks, participants used various retrieval plaforms, with a range of underlying document ranking models (Macdonald et al. 2008, 2009). We base our ranking methods on probabilistic, generative language models. Here, documents are ranked by the probability of the query being observed during randomly sampling words from the document. Since their introduction to the area of information retrieval, language modeling techniques have attracted a lot of attention (Hiemstra 2001, Miller et al. 1999, Ponte and Croft 1998). They are attractive because of their foundations in statistical theory, the great deal of complementary work on language modeling in speech recognition and natural language processing, and the fact that very simple language modeling retrieval methods have performed quite well empirically.

Work on blog feed search shows great resemblance to expert finding: given a topic, identify people that are experts on the topic. Our approach to the blog feed search task is modeled after two well-known language modeling-based models from the expert finding literature. In particular, our Blogger model corresponds to Model 1 in (Balog et al. 2006, 2009), while our Posting model corresponds to their Model 2. These connections were first detailed in (Balog et al. 2008, Weerkamp et al. 2008) and are examined and compared in great detail in this paper.

3 Probabilistic models for blog feed search

In this section we introduce two 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 and 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:

$$P(blog|q) = \frac{P(q|blog) \cdot P(blog)}{P(q)}, \tag{1}$$

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, $P(blog)$:

$$P(blog|q) \propto P(q|blog) \cdot P(blog). \tag{2}$$

Since we focus on a post-based approach to blog distillation, we assume the prior probability of a blog $P(blog)$ to be uniform. The distillation 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 two different ways. In our first model, the Blogger model (Sect. 3.1), 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 (Sect. 3.2), first retrieves individual blog posts that are relevant to the query, and then considers the blogs from which these posts originate.

The Blogger model and Posting model originate from the field of expert finding and correspond to Model 1 and Model 2 (Balog et al. 2006, 2009). 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 paper we use the open source implementation of both the Blogger and Posting model, called EARS:[6] Entity and Association Retrieval System.

3.1 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:

$$P(q|\theta_{blogger}(blog)) = \prod_{t \in q} P(t|\theta_{blogger}(blog))^{n(t,q)}, \tag{3}$$

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 $P(t)$:

$$P(t|\theta_{blogger}(blog)) = (1 - \lambda_{blog}) \cdot P(t|blog) + \lambda_{blog} \cdot P(t). \tag{4}$$

In Eq. 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:

$$P(t|blog) = \sum_{post \in blog} P(t|post, blog) \cdot P(post|blog). \tag{5}$$

We assume that terms are conditionally independent from the blog (given a post), thus $P(t|post, blog) = P(t|post)$. We approximate $P(t|post)$ with the standard maximum

[6] http://code.google.com/p/ears.

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 Sect. 6 we explore other ways of estimating this probability.

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.:

$$|blog| = \sum_{post \in blog} |post| \cdot P(post|blog), \tag{6}$$

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 (Mackay and Peto 1994). We set β to be the average blog length in the collection; see Table 4 for the actual values used in our experiments.

3.2 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 ($P(q|\theta_{posting}(post))$), weighted by their relative importance given the blog ($P(post|blog)$). Formally, this can be expressed as:

$$P(q|blog) = \sum_{post \in blog} P(q|\theta_{posting}(post)) \cdot P(post|blog). \tag{7}$$

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

$$P(q|\theta_{posting}(post)) = \prod_{t \in q} P(t|\theta_{posting}(post))^{n(t,q)}. \tag{8}$$

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

$$P(t|\theta_{posting}(post)) = (1 - \lambda_{post}) \cdot P(t|post) + \lambda_{post} \cdot P(t), \tag{9}$$

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 4 for the actual numbers used in our experiments.

3.3 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 Eq. 8 to retrieve blog posts that match a given query and construct a truncated list B of blogs these posts belong to. We do not need to "store" the ranking of this stage.

Stage 2: Given the list of blogs B, we use Eq. 3 to rank just the blogs that are present in this list.

By limiting the list of blogs B, in stage 1, 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 $(f \restriction N)(blog)$ 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 Eq. 8. Then,

$$P(q|\theta_{two}(blog)) = \begin{cases} 0, & \text{if } (f_q \restriction N)(blog) = \emptyset \\ \prod_{t \in q} P(t|\theta_{two}(blog))^{n(t,q)}, & \text{otherwise,} \end{cases} \tag{10}$$

where $(f_q \restriction N)(blog)$ denotes the set of top N relevant posts given the query and $\theta_{two}(blog)$ is defined as a mixture, just like Eq. 4:

$$P(t|\theta_{two}(blog)) = (1 - \lambda_{blog}) \cdot P_{two}(t|blog) + \lambda_{blog} \cdot P(t), \tag{11}$$

in which the key ingredient $P_{two}(t|blog)$ is defined as a variation on Eq. 5, restricted to the top M posts of the blog:

$$P_{two}(t|blog) = \sum_{post \in (f \restriction M)(blog)} P(t|post) \cdot P(post|blog). \tag{12}$$

Before examining the impact of the parameters N and M in Eqs. 10 and 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.

4 Experimental setup

We use the test sets made available by the TREC 2007 and 2008 blog tracks for the blog feed search task. Those collections consist of (1) a task definition, (2) a document collection, (3) a set of test topics, (4) relevance judgments ("ground truth"), and (5) evaluation metrics. Below, we detail those as well as statistics on our indexes and smoothing parameter β.

4.1 Document collection

The experiments presented in this paper use the TREC Blog06 collection (Macdonald and Ounis 2006). Table 1 lists the original collection statistics. The collection comes with three document types: (1) feeds, (2) permalinks, and (3) homepages. For our experiments, we only use the permalinks, that is, the HTML version of a blog post. During preprocessing, we removed the HTML code, and kept only the page title, and block level elements longer than 15 words, as detailed in (Hofmann and Weerkamp 2008).

Table 1 Statistics of the TRECBlog06 collection

Number of blogs	100,649
Number of posts	3,215,171
Uncompressed post size	88.8 GB
Crawl period	06/12/2005–21/02/2006

Table 2 Statistics of the 2007 and 2008 topic sets

	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

4.2 Topic sets

We use two predefined sets of test topics, which have been created during TREC 2007 and TREC 2008. The test topics follow the standard TREC format, consisting of a title (a few keywords), a description (a few sentences on what the topic is), and a narrative (a short story on which documents should be considered relevant and which ones should not). For our experiments, we are only interested in the title of a topic, which is comparable to a query submitted to a search engine by an end user. For each topic, relevance assessments are available: we know which blogs are relevant to the topic and which are not. If a blog has not been assessed for a given topic, the blog is considered to be non-relevant.

Looking at the relevance assessments for the 2007 and 2008 topics, we notice a few differences. Table 2 lists the statistics of the topics and relevance assessments for both years, while Fig. 1a 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.

We see that the 2008 topics have fewer relevant blogs per topic than the 2007 topics. Besides, looking at Fig. 1a and the last four lines in Table 2, 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 (>130 relevant blogs). These differences seem to be an artifact of the topic development guidelines[7,8] used in both 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.

[7] http://ir.dcs.gla.ac.uk/wiki/TREC-BLOG/TREC2007.

[8] http://ir.dcs.gla.ac.uk/wiki/TREC-BLOG/TREC2008.

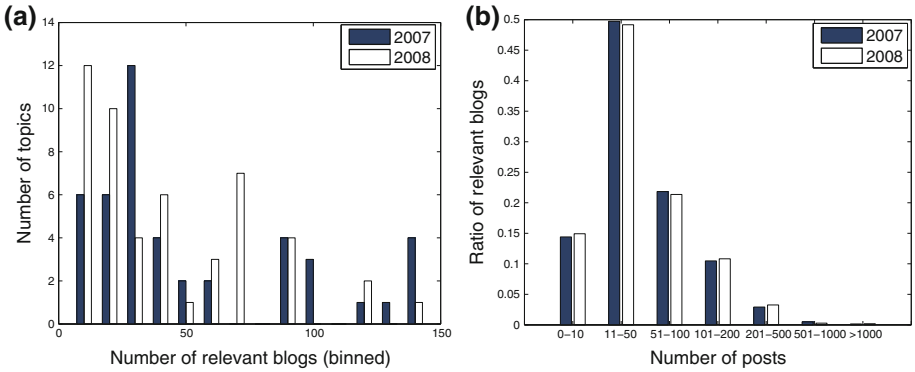

Fig. 1 (**a**) Number of relevant blogs (binned, *x* axis) vs number of topics with that number of relevant blogs (*y* axis). (**b**) Ratio of relevant blogs (*y* axis) with a certain size, in number of posts (*x* axis) for both 2007 and 2008 topics

We also look at the size of relevant blogs, in terms of the number of posts in a blog. In Fig. 1b 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.

4.3 Inverted indexes

We index the collection using the open source software package Lemur[9] (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 constructed using just the blog post title, creating a very lean index of the collection. Index statistics are listed in Table 3.

4.4 Smoothing

As explained in Sect. 3, 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. 6), and we list the values of β actually used in the paper in Table 4. We test the sensitivity of our models to the smoothing parameter β in Sect. 7.3.

4.5 Evaluation metrics and significance testing

To measure effectiveness, we report on three common IR metrics (Manning et al. 2008): a measure to capture precision and recall—mean average precision (MAP)—as well as two precision-oriented measures—precision at rank 5 (P@5), and mean reciprocal rank (MRR).

To test for significant differences between runs, we use a two-tailed paired t-test; $^{\blacktriangle}$ and $^{\blacktriangledown}$ reflect significant changes for $\alpha = .01$ and $^{\triangle}$ and $^{\triangledown}$ do the same for $\alpha = .05$.

[9] http://www.lemurproject.com.

Table 3 Statistics of the full content and title-only indexes

	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 4 Value of the smoothing parameter β for various runs of the Blogger and Posting model

Run		β (Blogger)	β (Posting)
All posts	Section 5.3	686	550
English posts	Section 5.3	630	506
English, no 1-post	Section 5.3	573	506
English, no 1-post, titles	Section 6.5	12	15
Comments, 50 posts	Section 6.3	595	–
Centrality, 50 posts	Section 6.3	590	–
Date, 50 posts	Section 6.3	575	–
Length, 50 posts	Section 6.3	615	–
Top 5,000 posts	Section 6.3	–	506

5 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 paper.

5.1 Language detection

The blog collection we use is a sample from the web (see Sect. 4.1) 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:[10] from 3,215,171 posts we remove 640,815 posts that are labeled as non-English, leaving us with 2,574,356 posts.

5.2 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

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

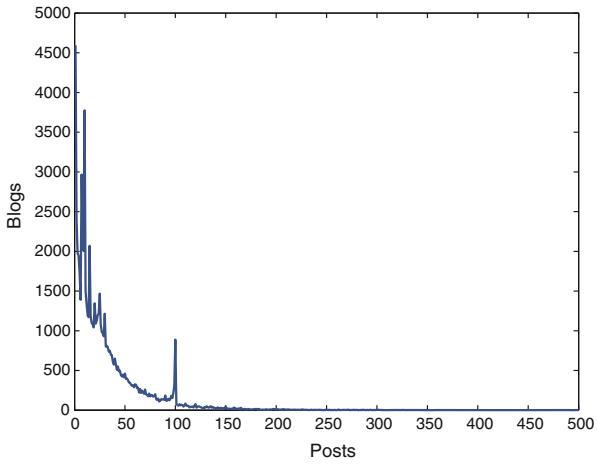

Fig. 2 Number of posts per blog

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."

In Fig. 2 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 artefact of the collection construction (see Sect. 4.1). A considerable number of blogs, 4,595 ($\sim$4%), consists of a single post. We do not want to exclude too many blogs, and therefore set $K = 1$, only dropping these 4,595 blogs from the index.

5.3 Baseline results

In Table 5 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, 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 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, 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.

Table 5 Baselines plus results of index pruning

Which posts?	2007			2008		
	MAP	P@5	MRR	MAP	P@5	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^{∇}	0.4360	0.7225
English only	0.3002	0.5067	0.6877	$0.2226^{\blacktriangledown}$	0.4160^{∇}	0.7021
English only, no short blogs	0.3140	0.5378	0.7055	0.2305^{∇}	0.4360	0.7237

Significance tested against Blogger model with all posts (top row)

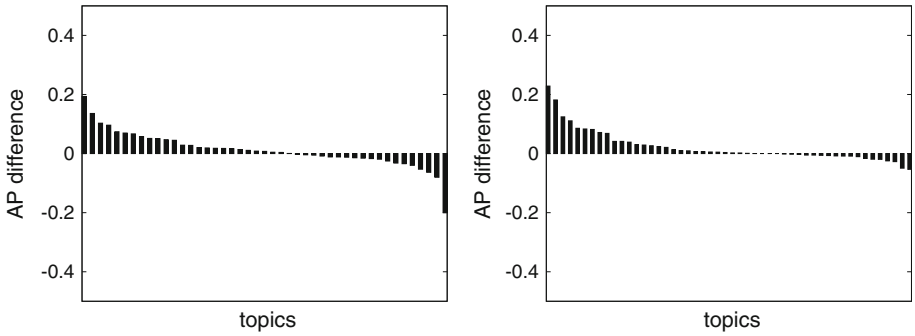

Fig. 3 Per-topic comparison on average precision for (*Left*) 2007 and (*Right*) 2008 topics for the Posting model (*baseline*) and the Blogger model

5.4 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 baseline. We look at both models using the pruned index (after removal of non-English posts and short blogs). Figure 3 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 6 summarizes the number of topics that prefer the Blogger model and the number of topics that prefer the Posting model.

Looking at which topics show very different performance in AP on both models, we find the topics displayed in Table 7. The results in Table 7 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 Fig. 4. We see that, on

Table 6 Number of topics that either prefer the Blogger model or 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 7 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.)

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

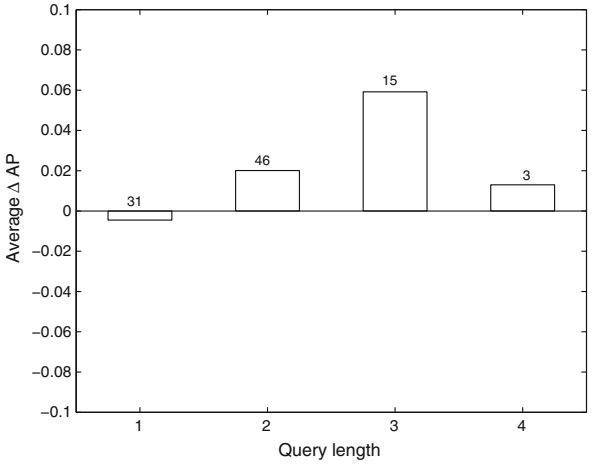

Fig. 4 Average improvement in AP for the Blogger model over the Posting model, grouped by query length. The number above the columns indicate the number of topics of that length

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 posts, we count the number of unique retrieved, relevant blogs for each model over the whole set of topics. Table 8 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.

Table 8 The number of unique relevant blogs for the Blogger and Posting model in the top 100 results

Model	2007	2008
Blogger	100	96
Posting	76	57

Table 9 The average size (in posts) of unique relevant blogs for both models

Model	2007	2008
Blogger	52	56
Posting	37	43

We explore these uniquely retrieved blogs in more detail and look at the size of the blogs (viz. Sect. 4.2), and list results in Table 9.

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

5.5 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 (Macdonald et al. 2008, 2009). 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.

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 10.

Table 10 Statistics of full content indexes used in the paper

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

6 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 Sect. 5 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 looking 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 Sect. 3.3. 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.

6.1 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). 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. 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. 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 11 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.

6.2 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) w.r.t. their importance to their parent blog; cut-offs as implemented in using the restricted summation in Eq. 12 will be based on these importance

Table 11 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

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**

rankings. Estimating post importance in blogs should ideally make use of blog specific features. In the following paragraphs we introduce three blog-specific features.

6.2.1 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. 13:

$$P(post|blog) = \frac{\log(|post|)}{\sum_{post' \in blog} \log(|post'|)} \tag{13}$$

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

6.2.2 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. 14).

$$KL(post||blog) = \sum_{t} P(t|post) \cdot \frac{P(t|post)}{P(t|blog)}. \tag{14}$$

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:

$$P(post|blog) = \frac{KL(post||blog)^{-1}}{\sum_{post' \in blog} KL(post'||blog)^{-1}}. \tag{15}$$

6.2.3 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. 16:

$$P(post|blog) = \frac{1 + \log(|comm(post)| + 1)}{\sum_{post' \in blog}(1 + \log(|comm(post')| + 1))}, \tag{16}$$

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.

6.3 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 Eq. 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 Eq. 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.

6.3.1 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 shows the performance in terms of MAP for the various ways of ordering and for multiple values of N.

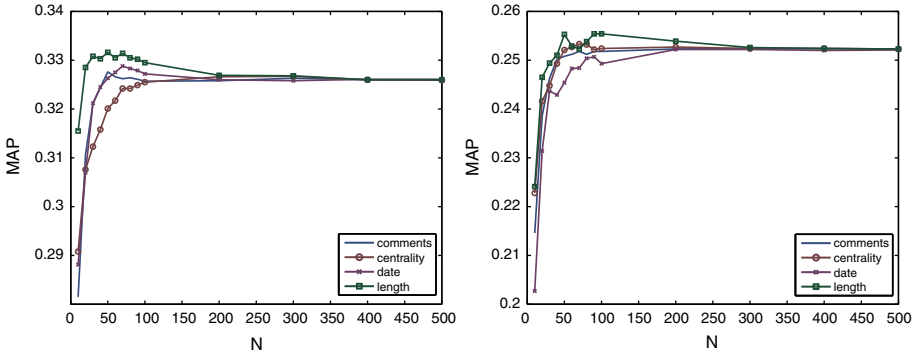

Fig. 5 Influence of selecting at most N posts on MAP of the Blogger model for (*Left*) 2007 and (*Right*) 2008, where posts are ordered by recency, comments, centrality, or length

Table 12 Number of associations that need to be considered when up to N posts are used for creating a Blogger model (regardless of ordering)

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

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

Table 13 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 all orderings improve effectiveness for at least some of the metrics, but *length* shows best overall improvements.

6.3.2 *Posting model*

Next we explore the impact of pruning on the effectiveness of the Posting model. In Fig. 6 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 14 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

Table 13 Results on the blog feed search task of the Blogger model built using at most top 50 posts, under various orderings

Ordering	2007			2008		
	MAP	P@5	MRR	MAP	P@5	MRR
– (All posts)	0.3260	0.5422	0.7193	0.2521	0.4880	0.7447
Recency	0.3263	**0.5600**	0.7110	0.2454^{∇}	0.4840	0.7423
Centrality	0.3201^{∇}	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**

Significance tested against all posts (top row)

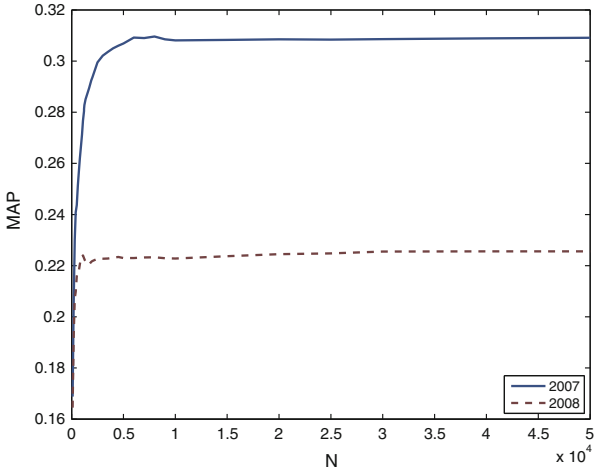

Fig. 6 Impact of limiting to the top N posts on MAP of the Posting model

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. 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 1 year as the training set and the other as the test set and averaging the optimal outcomes.

6.4 Evaluating the two-stage model

We quickly turn to the results achieved by the two-stage model as defined in Sect. 3.3. Table 15 lists the results of four settings, three of which we have already discussed: (1) the Blogger model (all posts), (2) the Blogger model with 50 posts (length ordered), (3) the Posting model with 5,000 posts, and (4) the two-stage model using items (2), and (3) as components (that is, with $N = 5{,}000$ and $M = 50$ in Eqs. 10 and 12, respectively).

Table 14 Results on the blog feed search task of the Posting model, with pruning, selecting only the top *N* posts

N	2007			2008		
	MAP	P@5	MRR	MAP	P@5	MRR
2,569,761 (all)	**0.3140**	**0.5378**	**0.7055**	**0.2305**	0.4360	0.7237
10,000	$0.3081^{\blacktriangledown}$	0.5244	0.6907	$0.2228^{\blacktriangledown}$	0.4360	0.7229
5,000	$0.3069^{\triangledown}$	0.5289	0.6912	$0.2230^{\blacktriangledown}$	0.4320	0.7232
1,000	$0.2712^{\blacktriangledown}$	0.5156	0.6821	0.2232	**0.4440**	**0.7403**
100	$0.1688^{\blacktriangledown}$	$0.4489^{\blacktriangledown}$	0.6729	$0.1645^{\blacktriangledown}$	0.4120	0.6980

Significance tested against the all posts runs (top row)

Table 15 Results on the blog feed search task of the combined approach

Setting	2007			2008		
	MAP	P@5	MRR	MAP	P@5	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)	$0.3069^{\triangledown}$	0.5289	0.6912	$0.2230^{\blacktriangledown}$	0.4320	0.7232
Two-stage model	**0.3334**	**0.5467**	**0.7321**	**0.2566**	**0.5040**	**0.7665**

Significance tested against the baseline (i.e., top row)

The results show that our two-stage model not only improves efficiency over the Blogger model, but it also leads to an increase in effectiveness. For the both topic sets we achieve the best performance in terms of all metrics using the two-stage model.

6.5 A further reduction

In Sect. 4.3 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: (1) full content for both stages, (2) title-only for the Posting model (stage 1), full content for the Blogger model (stage 2), (3) full content for the Posting model (stage 1), title-only for the Blogger model (stage 2), and (4) title-only in both stages. The results of these combinations are displayed in Table 16. For the 2007 topics the run using a title-only representation in stage 1, and the full content in stage 2 performs best on MAP, P@5, and MRR; the 2008 topics show a slightly mixed result, with full content representations in both stages performing best on MAP and P@5. What do these results mean? Using a lean title-only document representation in stage 1, the Posting model, is 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.

6.6 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 16 for the average results). The plots are given in Fig. 7.

Table 16 Results on the blog feed search task of different document representations in the two-stage model

Stage 1 (Posting)	Stage 2 (Blogger)	2007			2008		
		MAP	P@5	MRR	MAP	P@5	MRR
Full	Full	0.3334	0.5467	0.7321	**0.2566**	**0.5040**	0.7665
Title	Full	**0.3556**	**0.6533**$^{\blacktriangle}$	**0.8574**$^{\blacktriangle}$	0.2415	0.4840	**0.7794**
Full	Title	0.2719$^{\blacktriangledown}$	0.6178	0.7816	0.1995$^{\blacktriangledown}$	0.4776	0.7125
Title	Title	0.2601$^{\blacktriangledown}$	0.6133	0.7810	0.1889$^{\blacktriangledown}$	0.4640	0.6983

Significance tested against the best performing settings using full content for both stages (top row)

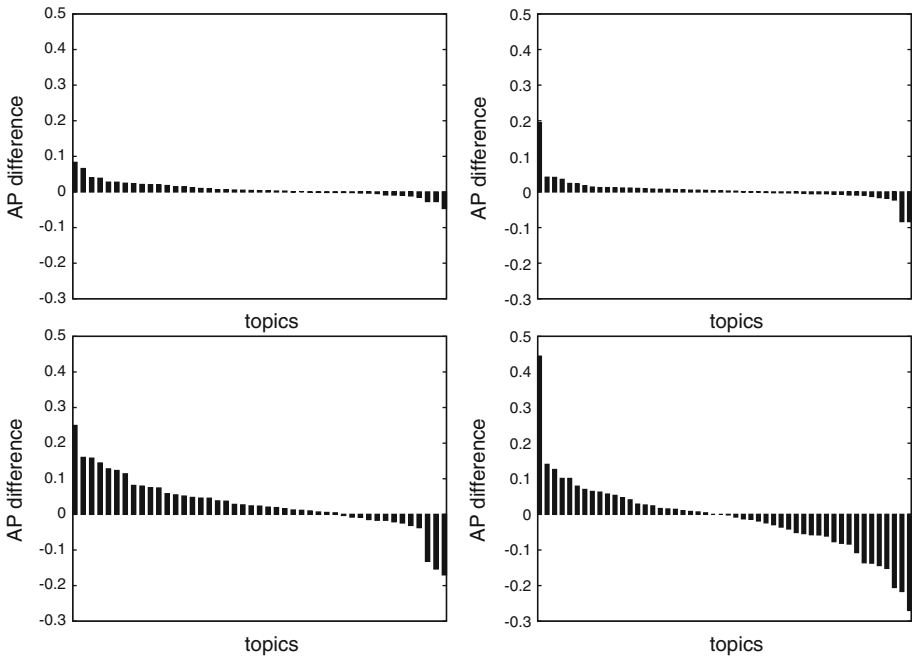

Fig. 7 Per-topic comparison for (*Left*) 2007 and (*Right*) 2008 topics on average precision (AP) for the baseline (Blogger model) compared to the two-stage model using (*Top*) full+full and (*Bottom*) title+full. *Positive bars* indicate better performance by the two-stage model, *negative bars* indicate better performance by the Blogger model

Table 17 Number of topics where performance goes "up" (⇑) or "down" (⇓) compared to the Blogger baseline

Run	2007						2008					
	AP		P@5		RR		AP		P@5		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 18 Topics that show an increase in performance on any metric going from the baseline to the two-stage model (title+full)

Topic	ΔAP	ΔP@5	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

The number in brackets is the topic ID

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 17. 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 P@5 and RR.

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 18 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 model. From the top improving topics, displayed in Table 18, 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 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 Fig. 8, where we see that (especially for 2008) the Blogger model retrieves more relevant blogs for most topics than the two-stage model.

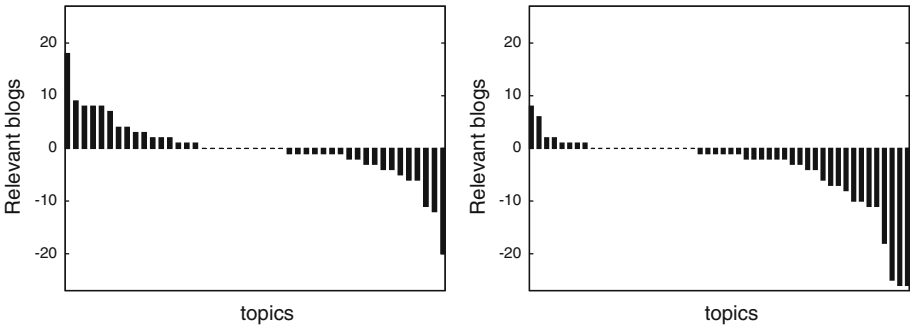

Fig. 8 Per-topic comparison for (*Left*) 2007 and (*Right*) 2008 topics on the number of relevant retrieved blogs for the baseline (Blogger model) and the combined model (title+full). *Positive bars* indicate more relevant results are retrieved by the two-stage model, *negative bars* indicate more relevant results are retrieved by the Blogger model

Table 19 The average size (in posts) of unique relevant blogs for both models

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

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 19 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 much larger (in terms of number of 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 Fig. 4. 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 Fig. 9 show that the two-stage model outperforms the Blogger model on one and two term queries, but shows a (very) slight decrease for longer queries.

6.7 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

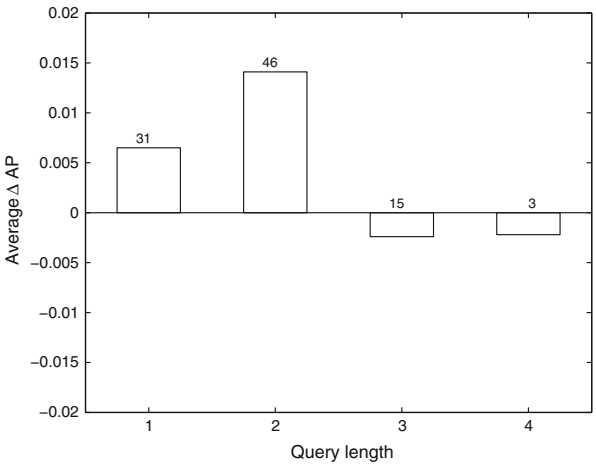

Fig. 9 Average improvement in AP for the two-stage model (title+full) over the Blogger model, grouped by query length. The number above the columns indicate the number of topics of that length

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 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 (Fig. 4); 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."

7 Discussion

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

7.1 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 Sect. 4. 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 20 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. The efficiency increase here 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.

Table 20 Efficiency vs. effectiveness for the Blogger model, Posting model, and the two-stage model

Model	Posts	2007			2008		
		MAP	P@5	MRR	MAP	P@5	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 21 Number of topics grouped by the rank of the first relevant result

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

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.

7.2 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 Sect. 6 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 21. 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.

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 (Weerkamp et al. 2009) (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.

7.3 Smoothing parameter

In Sect. 4.4 we briefly discussed the setting of 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 (Zhai and Lafferty 2004). To give an impression of this impact we run a baseline experiment for our two models (comparable to the "All" runs in Sect. 5.3). We compare the automatic setting of β (as detailed in Table 4) to a range of different β values (1, 10, 100, 1,000, 2,000, and 5,000) and list the results in Table 22.

Table 22 Impact of smoothing parameter β on effectiveness for the Blogger and the Posting model

β	2007			2008		
	MAP	P@5	MRR	MAP	P@5	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

Values corresponding to the automatic setting are typeset in italic

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 scores on the 2007 topic set for all retrieval metrics. On the 2008 set, a mixed picture emerges: best MAP and P@5 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.

8 Conclusions

In the paper 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.

A large part of the paper was devoted to pruning and representation techniques aimed at improving the efficiency of our models, especially that of the two-stage model, while maintaining (or even improving) effectiveness. Evaluations of the effectiveness of our models were performed using community-based benchmarking and complemented with topic-level analyses, as well as an analysis of the efficiency of our models as measured in terms of the number of core operations to be executed.

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 future work, we are interested in complementing our two-stage model with query modeling techniques as in (Elsas et al. 2008, Weerkamp et al. 2009), in going beyond the default implementations of the models to improve efficiency, in exploiting the ability of the two-stage model in ranking a relevant blog at position one (e.g., in extracting "relevant" characteristics), and in using different blog-specific features for ordering blog posts prior to pruning.

Acknowledgments We are grateful to our reviewers and the editors of the journal for providing valuable comments and feedback. This research was supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430 (GALATEAS), the 7th Framework Program of the European Commission, grant agreement no. 258191 (PROMISE), the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments under project nr STE-09-12, the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.815, 640.004.802, 380-70-011, the Center for Creation, Content and Technology (CCCT), the Hyperlocal Service Platform project funded by the Service Innovation & ICT program, and the WAHSP project funded by the CLARIN-nl program.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

Arguello, J., Elsas, J., Callan, J., & Carbonell, J. (2008). Document representation and query expansion models for blog recommendation. In ICWSM 2008.

Balog, K., Azzopardi, L., & de Rijke, M. (2006). Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2006)* (pp. 43–50). New York, NY, USA: ACM Press.

Balog, K., Azzopardi, L., & de Rijke, M. (2009). A language modeling framework for expert finding. *Information Processing and Management, 45*(1), 1–19.

Balog, K., de Rijke, M., & Weerkamp, W. (2008). Bloggers as experts: feed distillation using expert retrieval models. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2008)* (pp. 753–754). New York, NY, USA: ACM.

Eiron, N., & McCurley, K. S. (2003) Analysis of anchor text for web search. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR 2003)* (pp. 459–460).

Elsas, J., Arguello, J., Callan, J., & Carbonell, J. (2008). Retrieval and feedback models for blog distillation. In *The sixteenth text REtrieval conference proceedings (TREC 2007)*. NIST.

Elsas, J. L., Arguello, J., Callan, J., & Carbonell, J.G. (2008). Retrieval and feedback models for blog feed search. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 2008)* (pp. 347–354). New York, NY, USA: ACM.

Fujimura, K., Inoue, T., & Sugisaki, M. (2005). The eigenrumor algorithm for ranking blogs. In *WWW 2005 2nd annual workshop on the weblogging ecosystem: Aggregation, analysis and dynamics*. Chiba, Japan.

Fujimura, K., Toda, H., Inoue, T., Hiroshima, N., Kataoka, R., & Sugizaki, M. (2006). Blogranger—a multi-faceted blog search engine. In *Proceedings of the WWW 2006 3nd annual workshop on the weblogging ecosystem: Aggregation, analysis and dynamics*.

He, J., Weerkamp, W., Larson, M., & de Rijke, M. (2009). An effective coherence measure to determine topical consistency in user generated content. *International Journal on Document Analysis and Recognition, 12*(3), 185–203.

Hiemstra, D. (2001). Using language models for information retrieval. Ph.D. thesis, University of Twente.

Hofmann, K., & Weerkamp, W. (2008). Content extraction for information retrieval in blogs and intranets. Tech. rep., University of Amsterdam URL http://ilps.science.uva.nl/biblio/content-extraction-information-retrieval-blogs-and-intranets.

Jin, R., Hauptmann, A. G., & Zhai, C. X. (2002). Title language model for information retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 2002)* (pp. 42–48).

Lee, W. L., Lommatzsch, A., & Scheel, C. (2008). Feed distillation using adaboost and topic maps. In *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007)*. NIST.

Macdonald, C., & Ounis, I. (2006). The TREC Blogs06 collection: Creating and analysing a blog test collection. Tech. Rep. TR-2006-224, Department of Computer Science, University of Glasgow.

Macdonald, C., & Ounis, I. (2008). Key blog distillation: ranking aggregates. In *Proceeding of the 17th ACM Conference on Information and Knowledge Management (CIKM 2008)*, (pp. 1043–1052) New York, NY, USA: ACM.

Macdonald, C., Ounis, I., & Soboroff, I. (2008). Overview of the TREC 2007 Blog Track. In *The sixteenth text REtrieval conference proceedings (TREC 2007)*. NIST.

Macdonald, C., Ounis, I., & Soboroff, I. (2009). Overview of the TREC 2008 blog track. In *The seventeenth text REtrieval conference proceedings (TREC 2008)*. NIST.

Mackay, D. J. C., & Peto, L. (1994). A hierarchical dirichlet language model. *Natural Language Engineering, 1*(3), 1–19.

Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press.

Miller, D., Leek, T., Schwartz, R. (1999). A hidden Markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 1999)* (pp. 214–221).

Mishne, G. (2007) Applied text analytics for blogs. Ph.D. thesis, University of Amsterdam.

Mishne, G., & de Rijke, M. (2006). A study of blog search. In M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, & A. Yavlinsky (Eds) *Advances in information retrieval: Proceedings 28th European conference on IR research (ECIR 2006), LNCS* (Vol. 3936, pp. 289–301). New York: Springer.

Ounis, I., Macdonald, C., de Rijke, M., Mishne, G., & Soboroff, I. (2007). Overview of the TREC 2006 Blog Track. In *The fifteenth text retrieval conference (TREC 2006)*. NIST.

Ponte, J. M., & Croft, W. B. (1998) A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 2008)* (pp. 275–281). New York, NY, USA: ACM Press.

Seki, K., Kino, Y., Sato, S., & Uehara, K. (2007). TREC 2007 Blog Track Experiments at Kobe University. In *The sixteenth text REtrieval conference proceedings (TREC 2007)*. NIST.

Seo, J., & Croft, W. (2008). Blog site search using resource selection. In *Proceedings of the 17th ACM conference on information and knowledge management (CIKM 2008)* (pp. 1053–1062).

Seo, J., & Croft, W. B. (2008). UMass at TREC 2007 Blog distillation task. In *The sixteenth text REtrieval conference proceedings (TREC 2007)*. NIST.

Weerkamp, W., Balog, K., & de Rijke, M. (2008). Finding key bloggers, one post at a time. In *18th European conference on artificial intelligence (ECAI 2008)* (pp. 318–322). Patras, Greece.

Weerkamp, W., Balog, K., & de Rijke, M. (2009). A generative blog post retrieval model that uses query expansion based on external collections. In *Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP* (pp. 1057–1065). Association for Computational Linguistics, Singapore.

Weerkamp, W., & de Rijke, M. (2008). Credibility improves topical blog post retrieval. In *Proceedings of ACL-08: HLT* (pp. 923–931). Columbus, Ohio: Association for Computational Linguistics.

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