Blogger, Stick to your Story

Modeling Topical Noise in Blogs with Coherence Measures

Jiyin He j.he@uva.nl
Wouter Weerkamp w.weerkamp@uva.nl
Martha Larson m.a.larson@uva.nl
Maarten de Rijke mdr@science.uva.nl

ISLA, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam

ABSTRACT

Topical noise in blogs arises when bloggers digress from the central topical thrust of their blogs. We introduce a method to explicitly incorporate a model of topical noise into a language modeling approach to the task of blog distillation. Topical noise is integrated into the model using a coherence score, which reflects the tightness of the topical structure of a blog. Tests performed on the TRECBlog06 corpus show that a naive integration of the coherence score as blog prior fails to achieve performance improvements. Instead, we develop a set of more sophisticated models in which the coherence score is weighted by a function of the blog retrieval score. The proposed models help improve effectiveness of our language modeling approach to the blog distillation task.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.4 [**Information Systems Applications**]: H.4.2 Types of Systems; H.4.m Miscellaneous

General Terms

Algorithms, Measurement, Performance, Experimentation

Keywords

Coherence measures, blog distillation, language models

1. INTRODUCTION

The first years of the twenty-first century have witnessed a leap in the popularity of on-line journals supporting reader comments, otherwise known as weblogs or blogs [16]. New bloggers continue to enter the blogosphere, together generating an incredible number of new blog entries (posts) per day.[1] With this ever increasing amount of information available in the blogosphere, the need for intelligent access facilities grows as well.

Within the blogosphere we can identify different types of information needs: the need to find individual blog posts regarding a topic, or the need to identify blogs that frequently publish posts on a given topic. These information needs can be seen as related to query types identified by Mishne and de Rijke [17], where the distinction is made between ad hoc queries (short term interest) and filtering queries (long term interest). Although currently most focus is on finding blog posts, some systems like the Blogranger system [9] offer the possibility to retrieve full blogs, alongside of post-level retrieval functionality. Searchers can use blog search to identify blogs they would like to add to their feed readers.

The task we focus on is blog distillation: *identifying blogs that show a recurring interest in topic X*. The task has two main characteristics: first, the retrieval units are blogs rather than single posts; second, in order to be considered as relevant, a blog should not just mention the topic of the user query sporadically, but rather it must contain a significant number of posts concerning this topic. An effective approach to blog distillation should take both of these characteristics into account.

Two characteristics set blog content apart from conventional web content and necessitate that dedicated retrieval algorithms and approaches be developed for the blog domain. The first characteristic is the strong social aspect of blog content, most readily noticeable in the use of blog rolls, user assigned tags and especially comments to posts. The second characteristic, and the one most relevant to the current context, is the noisiness of the data in the blogosphere. We identify two levels on which blog content is noisy: (i) the blog post level and (ii) the blog level. At the post level, noisiness expresses itself in unexpected language usage, spelling and grammar errors, non-language characters (e.g., emoticons), and mixed data types (pictures, video, text). At the blog level, the noise can be characterized as *topical noise*. Topical noise tends to be semantic rather than lexical or structural. A blog, unlike, for example, the sports section of a newspaper, can (and most likely will) be about different topics. As a blogger, today I might write about the weather, tomorrow about soccer, and the day after about the earthquake in Sichuan. As an illustration of different levels of topical noise in blogs, consider Figures 1 and 2, two blogs treating the subject of vegetable gardening and displayed in the NetVibes[2] feedreader. In the blog in Figure 2, the blogger digresses from the topic of vegetable gardening to write about other topics. The topical freedom exercised by bloggers introduces topical noise, a type of noise yet to be dealt with extensively in the literature. Dealing effectively with topical noise is critical for improving performance on blog distillation, since blogs with topical noise show less consistent interest in particular subjects and are therefore a priori less likely to be appreciated by users in the setting of the blog distillation task.

The characteristics of the blog distillation task combined with the challenge presented by noisy data require a careful choice of an approach that is both flexible and sufficiently robust. Based on previous experiments we view blog distillation as an association finding task: which blogger is most closely associated with the given topic?

[1] Technorati reports over 1.5 million new blog posts every day.

[2] `http://www.netvibes.com`

AND'08, July 24, 2008, Singapore.

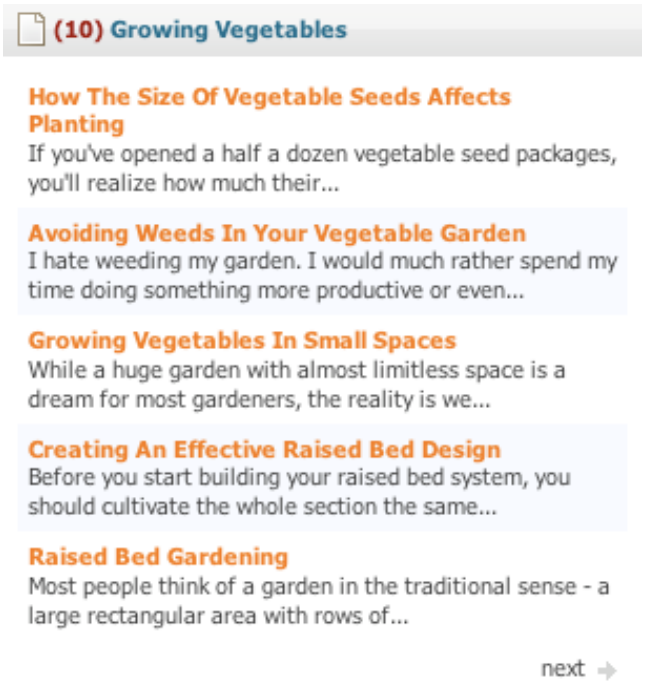


Figure 1: Example of blog with little to no topical noise

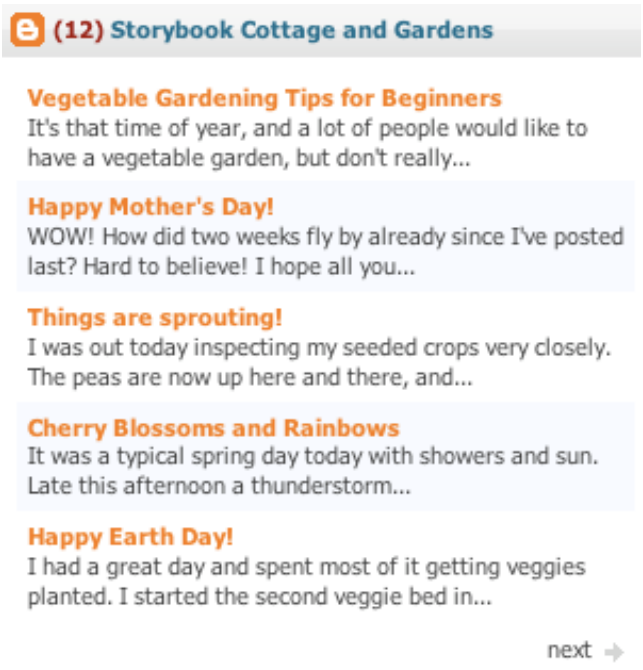


Figure 2: Example of blog with a moderate level of topical noise

And: how consistent is this blogger regarding the topic? To address the first question, we adopt the language modeling approach used in expert retrieval [3, 23]. To tackle the second question, we integrate a coherence score into the language modeling approach. The coherence score measures the topical clustering structure of a blog. Loose clustering reflects topical diffuseness and signals the presence of topical noise in the blog. Tight clustering indicates that the blog remains focused on one or a select few central themes. Given these two components, we explore the following dimensions in this paper: First, what is a proper way of estimating the coherence of a blog? Second, how can we use the coherence score in our retrieval process? Third, is the coherence of similar importance to all blogs? And finally, is there a minimum number of posts that need to be written in a blog before the coherence score has any influence?

Our main finding is that the proposed coherence score can help counter the topical noise present in blogs when it is weighted with the initial retrieval score, preventing blogs that display tight topic structure, but that are not relevant to the query, from rising to the top of the results list. Moreover, a substantial number of posts ($>$ 20) should have been written in a blog before the coherence score reaches its optimal performance increase.

The remainder of the paper is organized as follows: In Section 2 we discuss related work. Section 3 details our blog retrieval model and in Section 4 we introduce our coherence measure. Section 5 discusses the experimental setup and initial results. Section 6 presents and discusses the results of our analysis. In Section 7 we conclude and offer an outlook on future work.

2. RELATED WORK

In response to the growing interest in blogs and methods to access blog content, TREC launched a Blog track in 2006 [18]. The first year this track ran, its main focus was on identifying relevant and opinionated blog *posts* given a topic. Since the launch of this track, many new insights into blog post retrieval have been gained [14, 16, 18]. TREC 2007 introduced a new task in the Blog track: blog (or feed) distillation [14]. The aim is to return a ranking of blogs, rather than individual posts, given a topic; this is summarized as *find me a blog with a principal, recurring interest in X*. The scenario underlying this task is that of a user searching for feeds of blogs about a particular topic to add to a feed reader. This task is different from a filtering task [20] in which a user issues a repeating search on posts, constructing a feed from the results.

The main difference between the approaches applied by the different sites participating in TREC is the indexing unit used in the retrieval system: either full blogs [7, 22], or individual posts [7, 8, 22]. On top of either index, techniques like query expansion using Wikipedia [7] or topic maps [12] are applied. A particularly interesting approach—one that tries to capture the recurrence patterns of a blog—uses the notion of time and relevance [21]. After an initial retrieval run on a blog index, the relevance of all posts in the blog is determined and plotted against time. The area underneath this plot is considered to reflect the recurring interest of this blog for the given topic. Some additional techniques proved to be useful (e.g., query expansion), but most approaches did not lead to significant improvements over a baseline, or even led to a decrease in performance. Additionally, no approach attempted to explicitly incorporate the topical noisiness of blogs.

As mentioned above, topical noise in blogs arises when bloggers digress from the central topical thrust of their blogs. A blog with a high topical noise level contains posts on multiple topics and can be considered to be characterized by a relatively high degree of topical structure. Cluster analysis makes use of inter-document similarities and can be used to structure collections in topical groups. Clustering techniques have long been exploited for information retrieval purposes [24], in particular to allow users to browse and interact with collections in order to gain an understanding of their contents [1, 6]. Research making use of estimates of the number of topics present in a group of documents has been carried out in association with query performance prediction [4, 5]. Here, the number of topics contained in documents associated with a user query is used as an indicator of the ambiguity of the query, which in turn signals that a query can be expected to pose difficulty for a

retrieval system. Our recent work [10] has shown that coherence-based measures reflecting clustering structure can be used for the same purpose. The motivation behind using coherence scores is that they capture both the overall topical focus of a document set and the tightness of the clustering structure within that set. In the work reported here, we use a coherence score to reflect the topical structure of blogs in order to improve retrieval performance.

3. RETRIEVAL MODEL

Our approach to modeling topical noise in blogs builds on a retrieval model based on expert retrieval models [2]. As indexing unit we use individual blog posts. We have three reasons for this: (i) to allow for easy incremental indexing, (ii) for presentation of retrieval results posts are natural and coherent units, and (iii) the most important reason, to allow the use of one index for both blog post and blog retrieval [23].

We adopt a probabilistic approach to the task of determining relevance of blogs to the user query and formulate the task as follows: *what is the probability of a blog being relevant given the query topic q?* In other words, we estimate $p(blog|q)$, and rank blogs according to this probability. Since a query generally consists of only a few terms, often under-representing the information need that gave rise to it, Bayes' Theorem is applied in order to achieve a more accurate estimate:

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

where $p(blog)$ is the probability of a blog and $p(q)$ is the probability of a query. In Section 5, these two probabilities are described in greater detail; in the current section we focus on the estimation of the query likelihood, $p(q|blog)$: the likelihood of the topic expressed by the query q given a blog. Query likelihood estimation is accomplished using standard language modeling techniques. 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. The language modeling framework makes it possible to use blog posts to build associations between queries and blogs in a transparent and principled manner.

Our model represents a blog using a multinomial probability distribution over a vocabulary of terms. For each blog, a blog model θ_{blog} is inferred, such that the probability of a term given the blog model is $p(t|\theta_{blog})$. The model is then used to predict the likelihood that a blog gives rise to a particular query q. We make the assumption that each query term can be assumed to be sampled identically and independently from the blog model. Applying this assumption, the query likelihood is obtained by multiplying the likelihoods of the individual terms contained in the query:

$$p(q|\theta_{blog}) = \prod_{t \in q} p(t|\theta_{blog})^{n(t,q)}, \tag{2}$$

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

In order to prevent data sparseness from resulting in zero query likelihoods, we follow standard procedure and smooth the query likelihood model. The maximum likelihood estimate of the probability of a term given a blog $p(t|blog)$, which is then smoothed with term probabilities $p(t)$ estimated using the background collection:

$$p(t|\theta_{blog}) = \lambda_{blog} \cdot p(t|blog) + (1 - \lambda_{blog}) \cdot p(t). \tag{3}$$

In Eq. 3, $p(t)$ is the probability of a term in the document repository. The effect of smoothing is to add probability mass to the blog model in proportion to how likely that blog is to be generated (i.e., published) by a generic blogger.

The individual blog posts act as a bridge to connect t and the blog, resulting in the following estimate of $p(t|blog)$:

$$p(t|blog) = \sum_{post \in blog} p(t|post, blog) \cdot p(post|blog), \tag{4}$$

We make the assumption that the post and the blog are conditionally independent, setting $p(t|post, blog) = p(t|post)$. The importance of a given post within the blog is expressed by $p(post|blog)$. A simple approach to estimating this value is to assume a uniform distribution, i.e., all posts of a blog are weighted equally in terms of importance. Under this assumption, $p(post|blog) = posts(blog)^{-1}$, where $posts(blog)$ is the number of posts in the blog. Next, we present our method of estimating the smoothing parameter λ_{blog}.

3.1 Smoothing

The performance of language modeling-based retrieval methods is highly responsive to smoothing [25]. To estimate the smoothing parameter λ_{blog} in Eq. 3 in our model, we set λ_{blog} equal to $\frac{n(blog)}{\beta + n(blog)}$, where $n(blog)$ is the length of the blog (i.e., we sum the lengths of all posts of the blog). Essentially, the amount of smoothing applied to a given blog model is proportional to the length of that blog (and is like Bayes smoothing with a Dirichlet prior [15]). This approach is consistent with the observation that if a blog contains only few posts, estimation of the blog model is less robust and background probabilities are relatively more reliable and should thus make a larger contribution to the model. We set β to be the average blog length in the test collection (here, $\beta = 17,400$).

4. MEASURING COHERENCE

The coherence score we propose derives its inspiration from the *expression coherence* score used in the genetics literature [19]. An advantage of using a coherence score to measure topical structure is that it is not dependent on external linguistic resources such as thesauri.

A coherence score is a measure for the relative tightness of the clustering structure of a specific set of data as compared to the background collection. We use a coherence score in order to integrate information into our model about the level of topical noise present in a blog. In the current context, the data set consists of the blog posts within a blog, and the background collection is the full set of blog posts.

4.1 The coherence score

Given a set of documents $D = \{d_i\}_{i=1}^{M}$, which is drawn from a background collection C, i.e. $D \subseteq C$, we define the coherence score as the proportion of "coherent" pairs of document with respect to the total number of document pairs within D. The criterion of being a "coherent" pair is that the similarity between the two documents in the pair should meet or exceed a given threshold. Formally, given the document set D and a threshold τ, we have:

$$\delta(d_i, d_j) = \begin{cases} 1 & \text{if } \textit{similarity}(d_i, d_j) \geq \tau, \\ 0 & \text{otherwise.} \end{cases} \quad i \neq j \in \{1, \ldots, M\} \tag{5}$$

where the similarity between documents d_i and d_j can be any similarity metric. In this paper, we use the cosine similarity in our experiments, which is defined as:

$$cos(d_i, d_j) = \frac{d_i \cdot d_j}{\|d_i\| \|d_j\|} \tag{6}$$

We follow [10] and define the *coherence* (Co) of the document set D to be

$$Co(D) = \frac{\sum_{i \neq j \in \{1, \ldots, M\}} \delta(d_i, d_j)}{M(M-1)}. \tag{7}$$

From the above definition we can see that the threshold τ is an important parameter to determine the coherence score. As stated previously, the coherence score measures the relative tightness of the clustering structure of a set of documents compared to the background collection, the threshold τ actually establishes the connection between the two.

In order to obtain the value of τ, we randomly sample n documents from the background collection and calculate the pair-wise similarities. Then, we order the $\frac{1}{2}n(n-1)$ similarity scores and take the value of the score at the top $\alpha\%$ as the value of τ'. Heuristically we set $\alpha\%$ to 0.05, which means we assume that 5% pairs from the set of documents randomly drawn from the background collection are "coherent" pairs. We repeat this sampling for r runs and for different values of n and approximate the actually τ by taking the mean value of τ's from all these different runs. Any pairs of documents whose similarity meets or exceeds τ are considered to be "coherent" pairs.

4.2 Properties of the coherence score

The properties of the coherence score, and thereby its capacity to represent clustering structure, can be visualized by making use of a toy example. We generate four artificial data sets with different clustering structures. Data set (a) consists of one loose cluster which is generated by a normal distribution with large variance and we consider it to be the background set (or a random set). Data set (b) and (c) consist of 2 and 3 sub-clusters, respectively. Data set (d) consists of a tight cluster. Figure 3 illustrates these four data sets.

The variance is a commonly used measure for the "spreadness" of a data set: the smaller the variance, the more tightly the data points are gathered. We calculate the coherence score and the total variance for these four data sets. Table 1 shows the results. We can see that, ranking in terms of total variance, we have $(d) > (a) > (b) > (c)$; while in terms of coherence, we have $(d) > (b) > (c) > (a)$, whereby ">" means a tighter structure. The coherence score clearly surpasses the total variance in its ability to differentiate between data sets with and without clustering structure.

Table 1: The coherence score and the total variance of the toy data sets

datasets	(a)	(b)	(c)	(d)
coherence score	0.0006	0.0056	0.0034	0.0092
total variance	2.1227	2.1728	2.5315	0.1748

From this toy example, we can see that the coherence score prefers a structured data set to a random set, and among structured data sets, it prefers the sets with fewer sub-clusters. This property is crucial for the blog distillation task. When measuring the recurring interests in topic X, assuming all blogs are relevant to topic X, one would expect that the blogs with posts that concentrate on one topic to be more likely to have recurring interests, and that the blogs with posts that concentrate on limited number of topics to have at least some recurring interest in that topic.

5. EXPERIMENTS

In this section we introduce the collection and metrics we use, and we present the results of our initial experiments.

5.1 Test collection

The test corpus used in the experiments reported on here is the TRECBlog06 corpus [13]. The corpus was collected by monitoring feeds (blogs) for a period of 11 weeks and downloading .html documents behind all permalinks. For each permalink (or blog post or

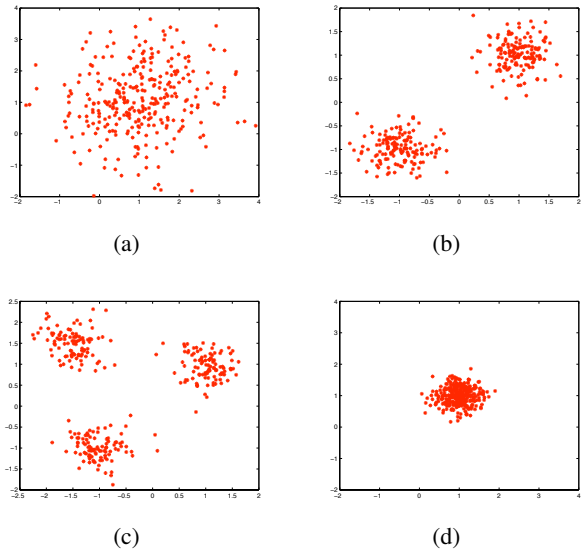


Figure 3: A toy example. (a) 1 random sample from a normal distribution with $\mu = (1, 1)$, $\sigma = 1$; (b) 2 random sample from a normal distribution with $\mu_1 = (-1, 1)$, $\mu_2 = (1, -1)$, $\sigma_1 = \sigma_2 = 0.5$; (c) 3 random sample from a normal distribution with $\mu_1 = (-1, 1)$, $\mu_2 = (1, -1)$, $\mu_3 = (-1.5, 1.5)$, $\sigma_1 = \sigma_2 = \sigma_3 = 0.5$; (d) 1 random sample from a normal distribution with $\mu = (1, 1)$, $\sigma = 0.3$.

document) the blog ID is registered. For these experiments we did not make use of the syndication information, which was available in addition to the .html content. The collection contains 3.2 million blog posts gathered from 100K blogs.

The TREC 2007 Blog track supplies 45 blog distillation topics, also referred to here as queries, and assessments concerning which blogs are relevant to which topics [14]. Topic development and assessment annotation were carried out by the participants of the track. In order to determine the relevance of a blog to a topic, assessors were asked to confirm that a substantial number of blog posts did indeed deal with that topic.

For all our runs we make use of the topic field (T) of the topics and discard the longer formulations of the topics (i.e., those contained in the description (D) and narrative (N) fields).

5.2 Metrics

In order to measure the performance of our approach to modeling topical noise in blog distillation, we use mean average precision (MAP) as well as three precision-oriented measures: precision at ranks 5 and 10 (P@5, P@10), and mean reciprocal rank (MRR).

5.3 Integrating the topical noise model

We extend our retrieval framework with the topical noise model by integrating the coherence score into our ranking function (Eq. 1). A transparent, straightforward integration can be implemented by taking the coherence score of a blog to supply information about query independent blog relevance, encoded in our model by the blog prior $p(blog)$. As detailed in Section 4, the coherence score is already a proportion, which means that it is scaled like a probability, and for this reason we can simply estimate $p(blog) = Co(blog)$, where $Co(blog)$ is calculated using Eq. 7. Since $p(q)$, the query probability, is equal for all documents we can drop this component for ranking purposes, and end up with the following retrieval score,

designated RSV (Retrieval Status Value):

$$RSV = p(q|blog)p(blog) \tag{8}$$

In cases where the coherence score of a blog is zero, or when no coherence can be calculated (in case of one-post blogs), we assign a low probability (.01). On one hand we do not want zero probabilities, but on the other hand we believe these blogs should not receive a high prior probability, since they do not show the recurring interest in a topic.

We compare the performance of the retrieval model with integrated coherence score to the performance of a baseline run. The baseline run uses uniform priors, i.e., we assign $p(blog) = |blogs|^{-1}$, where $|blogs|$ is the number of blogs in the collection.

The initial results of our baseline model, *baseline*, which uses a uniform prior, and our experimental model, *coherence*, which uses $Co(blog)$ (cf. Eq. 7) as the prior, are listed in Table 2. The run using coherence-based priors performs significantly worse in terms of MAP, but shows slight (non-significant) improvements on early precision (P@5) and MRR.

Run	MAP	P@5	P@10	MRR
baseline	.3272	.4844	.4844	.6892
coherence	.2945$^{\blacktriangledown}$	.5022	.4822	.6959

Table 2: Initial experiment results. Significant increase ($^{\blacktriangle}$) or decrease ($^{\blacktriangledown}$) is determined by a two-tailed paired t-test with $\alpha = .05$.

The question raised by the results of the experiments with this initial experimental model is the following: Why are the initial results (in terms of MAP) worse than the baseline? The fact that early precision and mean reciprocal rank (the ability to rank a relevant result on top) do show improvement suggests that the coherence score does indeed make a useful contribution in the case of top ranked blogs, but loses its ability to exert a positive effect when blogs appear further down the result list. Indeed, a (topically) relevant blog should not be highly favored in the relevance ranking unless it is also topically coherent. On the other hand, a blog that has high topical coherence because it consistently treats a different (non-relevant) topic might enjoy an unjustified promotion within the result list under the initial experimental model. Instead, we would like to target a more desirable behavior: top ranked blogs should enjoy a boost from the coherent score that allows them to maintain their prominence and bottom ranked blogs should be prevented from deriving benefit from their coherence score, since the chance is greater that they are coherent with respect to non-relevant topics. Documents in between should be given a moderate advantage if their coherence scores are high. In the next section, we propose a way of obtaining this behavior.

6. ANALYSIS

The model presented in the previous section failed to incorporate the coherence score in a way that improved retrieval performance. In this section, we first propose a more sophisticated method for integrating the coherence score, which makes use of estimated parameters. We report the outcomes of experiments aimed at testing the performance of the improved models. Finally, in Section 6.4 we introduce a length-related threshold that a blog must exceed in order to benefit from the effects of integrating the coherence score into the retrieval model.

6.1 Weighted coherence score

Based on the analysis offered at the end of the previous section we identify the following desired behavior: topically more relevant blogs should receive a solid boost if coherent, less relevant documents should not be affected. In fact, this boils down to weighting the coherence score by some notion of topical relevance. Since we do not have relevance judgements for our top ranked documents, we use the baseline retrieval score of a blog, i.e., Eq. 8, with a uniform prior, as a substitute for relevance.

Here, we prefer the retrieval score of the blog instead of using the rank of the blog in the retrieval result list, an obvious alternative. If the rank were used, a small difference in relevance could have a disproportionately large impact on the rank, making the weights over-sentive and unreliable.

In order to capture the desideratum that blogs with higher relevance receive bigger boosts from the coherence score, the weights are functions of RSV, the baseline retrieval score, and are designed to be monotonically increasing within the domain of RSV. In particular, we want blogs with RSV close to 0 to receive nearly no contribution from the coherence score while the blogs with highest RSVs receive the full impact from the coherence score, i.e., the range of the weights for the coherence scores should ideally be 0 to 1. The following functions modify the relation between the coherence weight ($W(\cdot)$) and the RSV in a manner consistent with this requirement. We have selected these functions to represent the range of possible relations between RSV and coherence score that we believe could potentially be useful.

- Linear function (*lin*):

$$W(RSV) = RSV \tag{9}$$

- Normal distribution with $\mu = 1$ and σ as a free parameter (*norm*):

$$W(RSV) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(RSV-\mu)^2}{2\sigma^2}\right) \tag{10}$$

- Quadratic function 1 (*quad1*):

$$W(RSV) = RSV^2 \tag{11}$$

- Quadratic function 2 (*quad2*):

$$W(RSV) = -(RSV-1)^2 + 1 \tag{12}$$

- Mixed function of 11 and 12 with α being the free parameter (*qmix*):

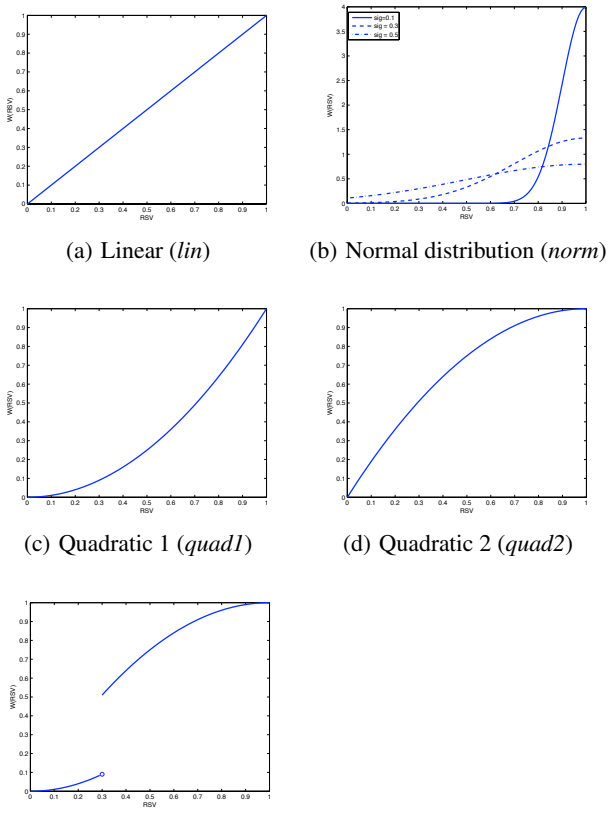
$$W(RSV) = \begin{cases} RSV^2 & \text{if } RSV < \alpha, \\ -(RSV-1)^2 + 1 & \text{otherwise.} \end{cases} \tag{13}$$

This choice allows us to explore a linear relation (Eq. 9), a non-linear relation with different rates of increase (Eq. 10, 11, 12), and combination of different rates of increase (Eq. 13). Figure 4 shows the curves of these functions in order to provide an intuition of the properties of the functions and their behaviors controlled by the values of the free parameters.

Finally, the weighted coherence score of a blog for a given query is defined as:

$$wCo(blog, query) = W(RSV) \cdot Co(blog) \tag{14}$$

The experimental models use wCo as the blog prior, substituting for $p(blog)$ in Eq. 1. We need one more refinement, however, as we will now discuss.



(a) Linear (*lin*) (b) Normal distribution (*norm*)

(c) Quadratic 1 (*quad1*) (d) Quadratic 2 (*quad2*)

(e) Quadratic mixed (*qmix*)

Figure 4: Weighting funstions

6.2 Normalization of RSV

As a common practice in language modeling approaches, when estimating the likelihood of a document given a query, i.e., Eq. 1, the denominator $p(q)$ is discarded because it does not affect the ranking of the results. However, when the impact of the coherence score is taken to be a function of the RSV, the normalization term is necessary in order to ensure that the weight of the coherence score is compatible across queries. A non-normalized RSV will impose an unwanted limitation of the domain and thereby also the range of the coherence score function.

In our experiments, we apply the full Bayes' Theorem, which leads to the estimation of the probability $p(q)$. To estimate $p(q)$ we adopt the method used by Lavrenko and Croft [11], in which the probability of a term $p(w)$ is estimated with the following equation:

$$p(w) = \sum_{m \in M} p(w|m)p(m), \tag{15}$$

where w is a term and M is a set of relevance models. We can translate this equation to our blog retrieval model by replacing $p(w)$ with $p(q)$ and M with B, a set of blogs. We end up with Eq. 16.

$$p(q) = \sum_{blog \in B} p(q|blog)p(blog) \tag{16}$$

We set B to be the top 200 results of query q so as to estimate $p(q)$.

6.2.1 Parameter estimation

For functions *norm* and *qmix* we need to set parameters σ and α. We performed a sweep over possible (and sensible) values of both parameters and evaluated the performance on MAP. The results of the sweeps are displayed in Figures 5 and 6, Based on which we select $\sigma = .05$ for *norm* and $\alpha = .05$ for *qmix*.

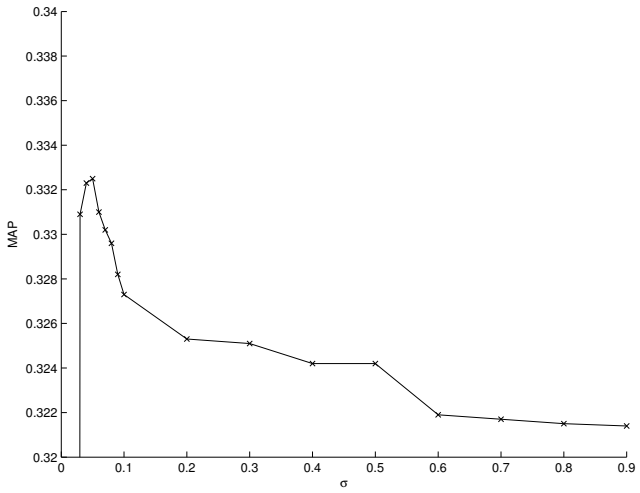


Figure 5: Effect of parameter σ on MAP for function *norm*.

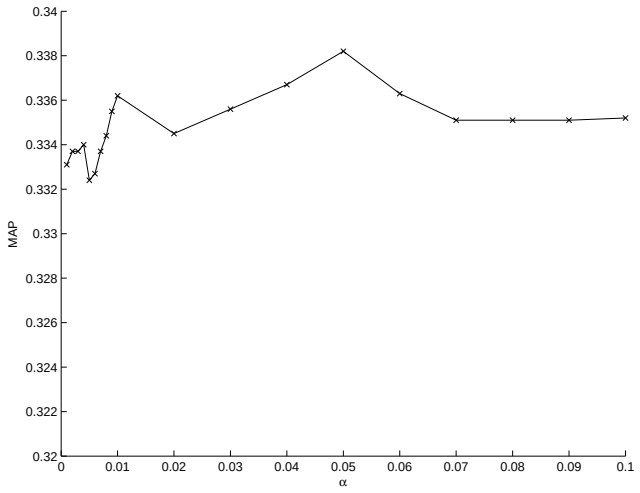


Figure 6: Effect of parameter α on MAP for function *qmix*.

Note that we are not trying to optimize the performance by selecting the best parameter, rather, we want to see the impact of the model parameter on the retrieval performance. For this reason, the generalization ability of the parameter setting is not considered. However, it would be interesting to test the optimized parameter settings on a separate collection and a separate set of queries.

6.3 Results of the retrieval model integrating the weighted coherence score

We generate runs using each of the five experimental models. These models are designated *lin*, *norm*, *quad1*, *quad2*, *qmix* according to which version of the weighted coherence score wCo they integrate. Each model ranks documents according to:

$$RSV = \frac{p(q|blog) \cdot wCo}{p(q)}, \tag{17}$$

The results for the different weighting functions are listed in Table 3. We can see that all functions show some improvement over

the baseline, with function *qmix* performing best in terms of MAP. The improvement gained over the baseline by applying this function as a weight to the coherence score is significant. We can see that the coherence score does not only help MAP, but also shows improvements on P@5, P@10 and MRR, although not significant.

Table 3: Results of weighted coherence score using linear function (*lin*), normal distribution (*norm*), quadratic function 1 (*quad1*), quadratic function 2 (*quad2*), and the combination of quadratic function 1 and 2 (*qmix*). Significance computed against baseline.

function	MAP	P@5	P@10	MRR
baseline	.3272	.4844	.4844	.6892
lin	.3326	.5022	**.5067**	.7266
norm	.3325	.5022	.4822	.7103
quad1	.3327	.5022	**.5067**	.7377
quad2	.3365	.5022	.5022	.7154
qmix	**.3382**▲	**.5067**	.5022	**.7394**

Let us take a closer look at the results per topic (i.e., query). In Figure 7 we compare the performance of each of the functions to the baseline and plot the increase or decrease in AP for each query. The plots show that (i) *norm* increases performance in 31 of 45 topics, but gains are moderate, (ii) function *quad1* hurts more topics than it improves (23 vs. 22), (iii) the same goes for *lin* (again 23 vs. 22), (iv) in both cases maximum increase in AP is high (.15 for topic 974), but maximum drop as well (-.14 for topic 979), (v) function *quad2* improves performance in 34 of 45 topics, but also shows a large drop for several topics, and finally (vi) function *qmix* improves over the baseline in 35 of 45 topics, with a limited drop in AP for the worst performing topic (-.07 for topic 979). The topic that improves most after integrating the coherence score into the model is topic 974, *tennis*, for all functions. Topic 979 has worst performance, *lighting*, for all functions. Topics whose performance neither improved or degraded include topic 951, *mutual funds*, topic 969, *planet*, and topic 933, *buffy vampire slayer*. We hypothesize that the potential of the coherence score to improve retrieval performance for a topic is related to the breadth of the vocabulary that a blogger uses to discuss the topic, the ability of the topic to inspire bloggers over time and the number of spam blogs whose word distributions cause them to be relevant to that topic. Further investigation of these hypotheses is left for future work.

We conclude this section with three additional remarks. First, in we discussed five functions to weight the coherence score. Many more can be used, but for this work it sufficed to show that weighting functions actually have a (positive) effect. Future work could be dedicated to proper ways of modeling the retrieval scores. Second, we used normal distribution as one of the weighting function here, but in fact we have tested other probability distributions as well, such as Cauchy distribution, Laplace distribution, etc. However, these probability distributions have a common problem for this task. That is, if we want to have more items in the ranking list to be affected by the coherence score, the weights each item received are decreased, since the total probability mass should sum up to 1. This constraint is unnecessary and empirically unfavorable. Third, we used a simple estimate of $p(q)$ in Eq. 16. More sophisticated ways of estimating this normalization factor could alter the influence of the coherence score and hence lead to better results.

6.4 Post threshold

To get a good estimation of the coherence of a set of documents this set needs to contain a certain number of documents; an estimation of coherence for a set of 3 documents is likely to be less

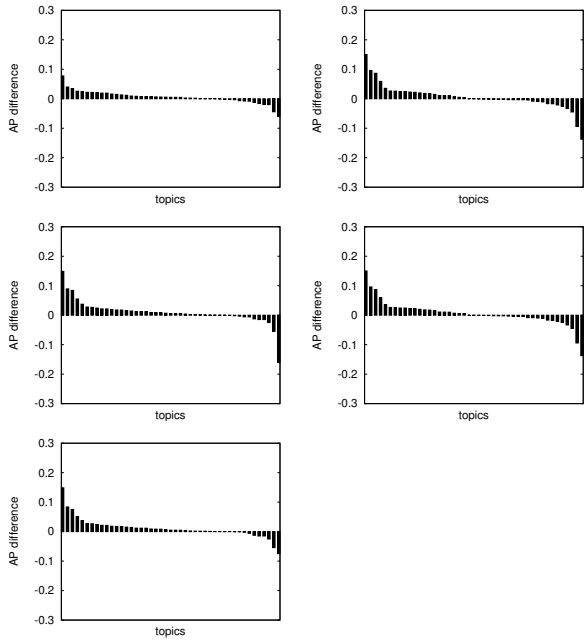


Figure 7: AP differences between baseline and (left-to-right, top-to-bottom) *norm*, *quad1*, *quad2*, *lin*, and *qmix*

accurate than an estimation for a set of 30 documents. In order to explore the implications of this limitation of the expressive ability of the coherence score, we carry out experiments to determine a post threshold that reflects how long a blog must be to derive benefit from the integration of the coherence score into the retrieval model. We experiment with thresholds between 0 and 50, and use the best performing settings for the five models (i.e. $\sigma = .05$ for *norm* and $\alpha = .05$ for *qmix*). Figure 8 plots the relative increase in MAP for each of the models over the baseline for different thresholds.

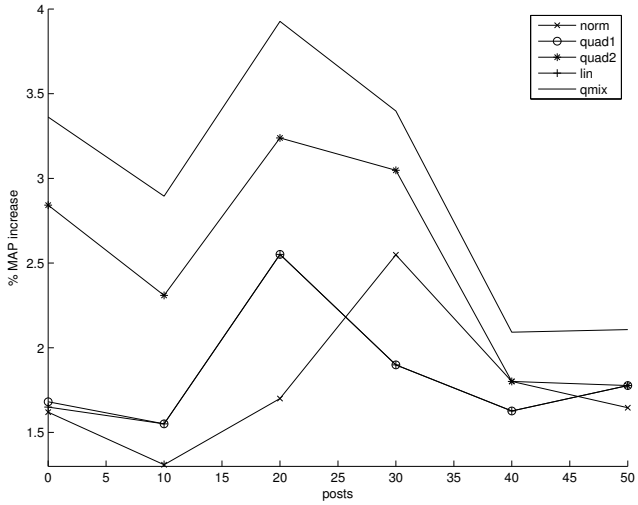


Figure 8: Effect of threshold on difference in MAP between models and baseline.

From the plot we can conclude that the greatest relative improvement over the baseline occurs when only blogs longer than 20 posts are taken into consideration. Only function *norm* has its peak at a threshold of 30 posts. Table 4 lists the results for each of the func-

tions and the baseline when using a threshold of 20 posts.

Table 4: Results of weighted coherence score applied to blogs with a minimum of 20 posts. Significance computed against baseline.

function	MAP	P@5	P@10	MRR
baseline	.2470	.4578	.4511	.6930
lin	.2533	.4756	.4689	.7174
norm	.2512▲	.4756	.4622	.7030
quad1	.2534	.4756	.4689	.7285
quad2	.2550▲	.4756	**.4711**	.7061
qmix	**.2567▲**	**.4800**	**.4711**	**.7321**

The results show that in three cases the improvement over the baseline is significant (in terms of MAP), and that, again, function *qmix* performs best on all metrics. The results suggests that it would be beneficial to develop methods to estimate priors for blogs that are too short to derive benefit from the coherence score. In the experiments reported in Table 3, we set the prior of blogs containing a single post to $p(blog) = .01$. We felt it would be unjustified to extend this assumption to blogs longer than one post but shorter than 20, since as a blog grows beyond a single post a recurrence pattern starts to emerge. What is unclear is the amount of growth necessary for stabilization. We believe that a suitable estimation of the blog prior for short blogs (< 20 posts) needs to be informed by a better understanding of user perceptions of the impact of topic recurrence in short blogs on relevance within the blog distillation tasks.

7. CONCLUSION

In this paper we proposed a method to counteract the effects of topical noise in blogs with the goal of performing blog distillation. For a blog to be relevant in a distillation task, it should show recurring interest in a given topic, something that is hard to measure due to the noisiness of blogs on a blog level.

To this end we introduced a coherence score, which captures the topical clustering structure of a set of documents as compared to the background collection. Applied to blogs, this score reflects topical consistency, in other words, the level of topical noise of a blog.

Initial results showed that implementing the coherence score as a document prior hurts performance significantly. Based on an analysis of the initial results, weighting the coherence score using a function dependent on the retrieval score is proposed; five different weighting functions are introduced and when these are used to weight the coherence scores, peformance improves over the baseline on all metrics.

Using the coherence score as indicator of topical consistency does not make sense for blogs of all lengths. Experiments with a threshold on the number of posts within a blog shows that maximum improvement over the baseline is obtained when looking at blogs with a minimum of 20 posts.

Detailed per-topic analysis shows that some weighting functions (*quad1* and *lin*) hurt more topics than they help in terms of AP, even though their MAP is slightly higher than the baseline. Functions *norm*, *quad2*, and *qmix* help in most cases (75%) and function *qmix* performs best on almost all metrics.

Future work will focus on a better approximation of the weighting function, more intelligent ways of estimating the normalization factor as well as the model parameters. Moreover, a close study on the query-specific performance for our method would also be interesting, as it is not yet clear if the influence of the weighting models are query dependent, i.e., queries of different characteristics may prefer different weighting functions.

8. ACKNOWLEDGEMENTS

This research was supported by the E.U. IST programme of the 6th FP for RTD under project MultiMATCH contract IST-033104, and by the Netherlands Organisation for Scientific Research (NWO) under project numbers 220-80-001, 017.001.190, 640.001.501, 640.-002.501, 612.066.512, STE-07-012.

9. REFERENCES

[1] R. B. Allen, P. Obry, and M. Littman. An interface for navigating clustered document sets returned by queries. In *COCS'93*, pages 166–171, New York, NY, USA, 1993. ACM.

[2] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR'06*, pages 43–50, New York, NY, USA, 2006. ACM Press.

[3] K. Balog, M. de Rijke, and W. Weerkamp. Bloggers as experts. In *SIGIR'08*, 2008.

[4] S. Cronen-Townsend and W. B. Croft. Quantifying query ambiguity. In *HLT'02*, pages 94–98, 2002.

[5] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *SIGIR'02*, pages 299–306, 2002.

[6] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. In *SIGIR '92*, pages 318–329, NY, USA, 1992. ACM.

[7] J. Elsas, J. Arguello, J. Callan, and J. Carbonell. Retrieval and feedback models for blog distillation. In *TREC'07 Working Notes*, 2007.

[8] B. J. Ernsting, W. Weerkamp, and M. de Rijke. The University of Amsterdam at the TREC 2007 Blog Track. In *TREC'07 Working Notes*, 2007.

[9] K. Fujimura, H. Toda, T. Inoue, N. Hiroshima, R. Kataoka, and M. Sugizaki. Blogranger—a multi-faceted blog search engine. In *WWW'06*, 2006.

[10] J. He, M. Larson, and M. de Rijke. Using coherence-based measures to predict query difficulty. In *ECIR'08*, pages 689–694, 2008.

[11] V. Lavrenko and W. B. Croft. Relevance-based language models. In *SIGIR '01*, pages 120–127, NY, USA, 2001. ACM Press.

[12] W.-L. Lee and A. Lommatzsch. Feed distillation using adaboost and topic maps. In *TREC '07 Working Notes*, 2007.

[13] C. Macdonald and I. Ounis. The TREC Blogs06 collection: Creating and analyzing a blog test collection. Technical Report TR-2006-224, Department of Computer Science, University of Glasgow, 2006.

[14] C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC 2007 Blog Track. In *TREC '07 Working Notes*, pages 31–43, 2007.

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

[16] G. Mishne. *Applied Text Analytics for Blogs*. PhD thesis, University of Amsterdam, 2007.

[17] G. Mishne and M. de Rijke. A study of blog search. In M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, editors, *ECIR'06*, volume 3936, pages 289–301, April 2006.

[18] I. Ounis, C. Macdonald, M. de Rijke, G. Mishne, and I. Soboroff. Overview of the TREC 2006 Blog Track. In *TREC '06*. NIST, 2007.

[19] Y. Pilpel, P. Sudarsanam, and G. M. Church. Identifying regulatory networks by combinatiorial analysis of promoter elements. *Nat. Genet.*, 29:153–159, 2001.

[20] S. Robertson and J. Callan. Routing and filtering. In *TREC '05*, pages 99–122. MIT, 2005.

[21] K. Seki, Y. Kino, and S. Sato. TREC 2007 Blog Track Experiments at Kobe University. In *TREC '07 Working Notes*, 2007.

[22] J. Seo and W. B. Croft. UMass at TREC 2007 Blog Distillation Task. In *TREC '07 Working Notes*, 2007.

[23] W. Weerkamp, K. Balog, and M. de Rijke. Finding key bloggers, one post at a time. In *ECAI'08*, 2008.

[24] P. Willett. Recent trends in hierarchic document clustering: a critical review. *Inf. Process. Manage.*, 24(5):577–597, 1988.

[25] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2): 179–214, 2004.