IJDAR
DOI 10.1007/s10032-009-0089-5

ORIGINAL PAPER

An effective coherence measure to determine topical consistency in user-generated content

Jiyin He · Wouter Weerkamp · Martha Larson · Maarten de Rijke

Received: 20 November 2008 / Revised: 19 May 2009 / Accepted: 9 July 2009

Abstract When searching for blogs on a specific topic, information seekers prefer blogs that place a central focus on that topic over blogs whose mention of the topic is diffuse or incidental. In order to present users with better blog feed search results, we developed a measure of topical consistency that is able to capture whether or not a blog is topically focused. The measure, called the *coherence score*, is inspired by the genetics literature and captures the tightness of the clustering structure of a data set relative to a background collection. In a set of experiments on synthetic data, the coherence score is shown to provide a faithful reflection of topic clustering structure. The properties that make the coherence score more appropriate than lexical cohesion, a common measure of topical structure, are discussed. Retrieval experiments show that integrating the coherence score as a prior in a language modeling-based approach to blog feed search improves retrieval effectiveness. The coherence score must, however, be used judiciously in order to avoid boosting the ranking of irrelevant but topically focused blogs. To this end, we experiment with a series of weighting schemes that adjust the contribution of the coherence score according to the relevance of a blog to the user query. An appropriate weighting scheme is able to improve retrieval performance. Finally, we show that the coherence score can be reliably estimated with a sample exceeding 20 posts in size. Consistent with this finding, experiments show that the best retrieval performance is achieved if coherence scores are used when a blog contains more than 20 posts.

Keywords User-generated content · Topical structure · Information retrieval · Blog search · Coherence

1 Introduction

The amount of user-generated content available on-line is already voluminous, and it continues to grow on a daily basis. User-generated content is not regulated by top-down rules, leaving users free to decide (i) what to write about (topics), (ii) how to write (writing style, language), and (iii) when to write (time of day, regularity). Since user-generated content is produced without editor supervision, standards and conventions that otherwise dictate the form and consistency of written prose cannot be assumed to be upheld. A specific type of user-generated content, blogs (syndicated web journals), has shown a particularly spectacular rise. Currently, bloggers generate content at a rate in the order of one million new posts per day.[1] With this ever increasing amount of information available in the blogosphere, the need for intelligent access facilities grows as well.

The information needs of users searching the blogosphere fall into two general categories: the need to find individual blog posts regarding a topic, or the need to identify blogs that

This paper is a revised and extended version of [19].

J. He (✉) · W. Weerkamp · M. de Rijke
ISLA, University of Amsterdam, Science Park 107,
1098GX Amsterdam, The Netherlands
e-mail: j.he@uva.nl

W. Weerkamp
e-mail: w.weerkamp@uva.nl

M. de Rijke
e-mail: mdr@science.uva.nl

M. Larson
EEMCS, Delft University of Technology, Mekelweg 4,
2628 CD Delft, The Netherlands
e-mail: m.a.larson@tudelft.nl

[1] http://technorati.com/blogging/state-of-the-blogosphere/. Accessed 19 May 2009.

Published online: 13 August 2009

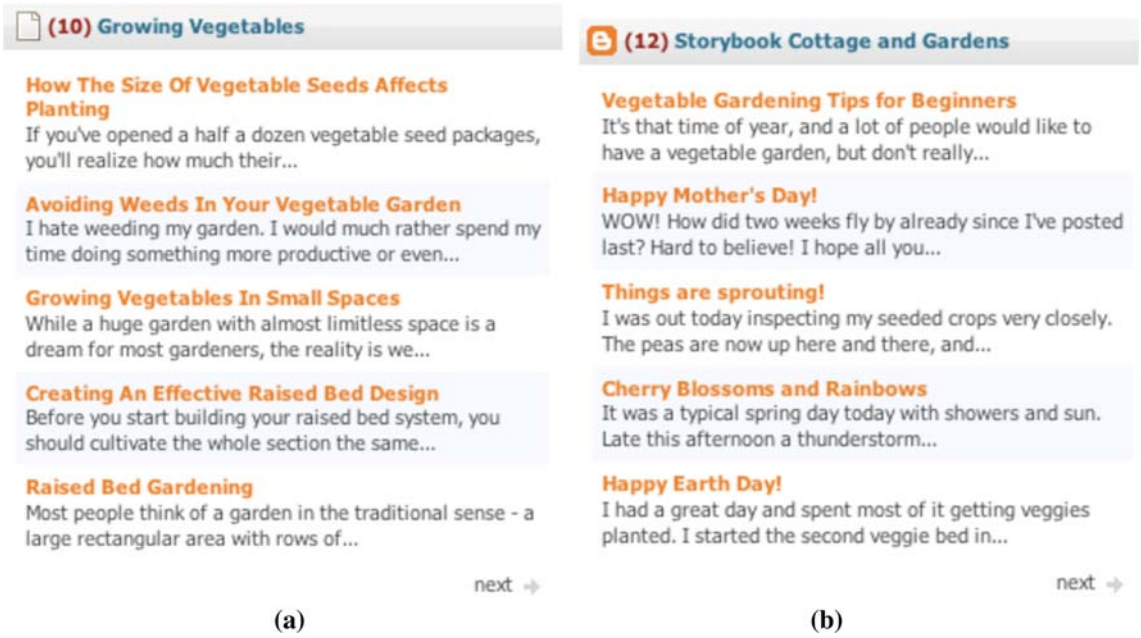

Fig. 1 (**a**) Example of a blog with little to no topical noise. (**b**) Example of a blog with a moderate level of topical noise

frequently publish posts on a given topic. These categories mirror the short term versus long term interest distinction observed by Mishne and de Rijke [27] in their study of blog search behavior. Although currently most focus is on finding blog posts, some systems offer the possibility to search for full blogs, alongside of post-level retrieval functionality [14]. Searchers can use blog search to identify blog feeds they would like to add to their feed readers.

The task we focus on in this paper is blog feed search, also called blog distillation [24]. The blog feed search task is defined as *identifying blogs that show a central, recurring interest in a given topic*. 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 feed search should take both of these characteristics into account.

Two key features set blog content apart from conventional web content and necessitate that dedicated retrieval algorithms and approaches be developed for the blog domain. The first 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, 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, noise 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*, which tends to be semantic rather than lexical or structural. A blog can (and most likely will) be about different topics. As an illustration of different levels of topical noise in blogs, consider Fig. 1a, b, two blogs treating the subject of vegetable gardening and displayed in the NetVibes[2] feedreader. In the blog in Fig. 1b, the blogger digresses from the topic of vegetable gardening to write about other topics. Dealing effectively with this type of topical noise is critical for improving performance on blog feed search, 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 feed search task.

How can we measure topical noise? Specifically, how can we measure it in blogs? The characteristics of the blog feed search task combined with the challenge presented by noisy data require an approach that is both flexible and sufficiently robust. We view blog feed search as an association finding task: which blogger is most closely associated with the given topic? And: how consistent is this blogger regarding the topic? To address the first issue, we adopt the language modeling approach used in expert retrieval [4,36]. To tackle the second issue—the core issue addressed in this paper—, we integrate a *coherence score* into this language modeling-based approach. The coherence score measures the topical clustering structure of a blog. Loose clustering structure reflects topical diffuseness and signals the presence of topi-

[2] http://www.netvibes.com. Accessed 19 May 2009.

cal noise in the blog. Tight clustering structure indicates that the blog remains focused on one or a few central themes.

Given these issues, we explore the following three dimensions in this paper, which we formulate as research questions: First, *what is a proper way of estimating the coherence of a blog?* Here, we offer our coherence score as a solution; we compare it against lexical cohesion, a standard measure for determining the diversity of topics discussed in a text. Second, *how can we use the coherence score in our retrieval process?* Here, we compare a number of options, ranging from treating the coherence score as a simple prior to modeling it as a multiplicative factor whose contribution is a function of the retrieval status value of a blog. Finally, and given that the collection we use in our experiments only hands us a sample of blog posts generated by the underlying blog models, *how does the sample size influence the estimation of the coherence and how does this influence blog feed search?* This final question is addressed using an experimental exploration.

Our main finding is that our proposed coherence score can estimate the topical noise present in blogs. Moreover, it 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. In addition, a minimum of 20 posts is required to get a proper estimate of the coherence of a blog, regardless of the actual size of the blog. This finding is supported by blog feed search results: The coherence score reaches its optimal performance increase when a substantial number of posts (>20) have been written in a blog.

The remainder of the paper is organized as follows: In Sect. 2 we discuss related work and introduce the collections we use throughout the paper. In Sect. 3 we introduce and study our proposed coherence score; we compare it against the well-known lexical cohesion measure. In Sect. 4 we detail the modeling of blog feed search and the integration of coherence in this framework. Section 5 specifies our experimental settings, and we discuss the results of the experiments in Sect. 6. In Sect. 7 we analyze our experimental findings, before concluding in Sect. 8.

2 Background

In this section we first give an overview of previous work related to the issues addressed in this paper. The second part introduces the collections we use throughout the paper.

2.1 Related work

As illustrated by our introductory examples, topical noise in blogs arises when bloggers treat a variety of topics and the blog fails to develop or maintain a central topical thrust. A blog with a high topical noise level contains posts on multiple topics and can be considered to be characterized by a relatively loose topical structure. Cluster analysis makes use of inter-document similarities and can be used to structure collections in topical groups. Estimating the number of clusters is a difficult problem and there is no absolute best way to find out the number of clusters [6,12,16]. On the other hand, it is not necessary that we know the exact number of clusters, rather, we are more interested in the distribution of the blog posts in the semantic space.

Clustering techniques have long been exploited for information retrieval purposes [39], in particular to allow users to browse and interact with collections in order to gain an understanding of their contents [1,9]. 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 [7,8]. 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 [18] has shown that coherence-based measures reflecting clustering structure implicitly 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 response to the growing interest in blogs and methods to access blog content, The text retrieval conference (TREC) launched a Blog track in 2006 [29]. 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 [24,26,29]. TREC 2007 introduced a new task in the Blog track: blog (or feed) distillation [24] (in our paper referred to as blog feed search). 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 central, recurring interest in a given topic*. 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 [31] 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 [10,33], or individual posts [10,11,33]. On top of either index, techniques like query expansion using Wikipedia [10] or topic maps [21] are applied. A particularly interesting approach—one that tries to capture the recurrence patterns of a blog—uses the notion of time and relevance [32]. 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.

The voting-model-based approach of [23] is competitive with the TREC-2007 blog feed search results reported in [24]. This approach shares with ours the ambition to exploit information about topical patterns in blogs to improve the results of blog feed search. The existence of three possible topical patterns is postulated and models are formulated that attempt to encode each into the blog retrieval model. As in [38], *central interest* is captured using a query-based cluster score designed to reflect the relevance of the central topic of the blog to the query. *Recurring interest* is captured using a query-based date score that breaks the temporal window of the data collection down into time-based intervals and sums a topical contribution from each interval. Tuning involves setting the optimal width of the time based interval. This approach resembles that taken in [11], which incorporates topical relevance from the most recent interval rather than from all intervals. *Focused interest* is captured by the integration of a score measuring the cohesiveness of the language models used in the set of posts in a blog.

Our work moves beyond the approach proposed in [23], eliminating the need to target individual topical patterns and to tune multiple topical-pattern-based scores. Our proposed use of the coherence score to encode the topical structure of blogs allows us to simultaneously capture both the topical focus at blog level and the tightness of the relatedness of sub-topics within the blog. In the work reported here, we use a coherence score to reflect the topical structure of blogs in order to improve retrieval performance. Our approach finds motivation in our previous successful use of the coherence score, [18], to represent topical structure. The conceptual simplicity of the coherence score is innately appealing. Its ability to implicitly capture clustering structure as a whole without tuning of individual parameters related to data partitioning, cluster number or cluster size makes the coherence score a particularly attractive means of integrating topical information into a system to improve blog feed search.

2.2 Collections

Before we dive into topical consistency measures and their influence on blog feed search, we first detail the collections we use throughout the paper. In Sect. 3.2.2 we apply the coherence score to three different collections. The retrieval experiments in Sect. 6 are done on one of these collections, Blog06.

The two TREC collections (collections used by the Text REtrieval Conference) we use besides the blog collection, are listed in Table 1 alongside their main document types and the topics that are used in the experiments in Sect. 3.2.2. Collections at TREC are used within the setting of a certain *track* (e.g., blog track or enterprise track) and *task* (e.g., opinionated blog post finding or expert finding). For each of these tasks, a set of topics is available; topics are composed of a keyword query, a description of the topic, and a narrative of the user information need. For each of the topics relevance judgments are available to allow participants to test their systems on the same set of topics and compare results to other systems.

Table 1 Collection characteristics and used topics

Collection	Document type	Topics
AP89+88	News	1–200
Robust04	News, governmental	301–450, 601–700
Blog06	Blog posts	851–950

The Blog06 corpus [22] 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 document) the blog ID is registered. For these experiments we did not make use of the syndication information (i.e., RSS feeds). The collection contains 3.2 million blog posts gathered from 100K blogs.

Our aim in selecting these particular collections is the different types of documents they contain: the blog collection contains individual blog posts, that is, user-generated content. The other two collections contain formal, edited content from two sources: news and governmental pages. User-generated content, with its lack of editors and top-down rules, differs from more formal, edited content in various ways, for example (i) spelling and grammatical errors are more common in blogs because of the lack of editors, (ii) language usage in blogs is more diverse, whereas formal content often uses a fairly narrow vocabulary. These collections allow us to check whether we can use one measure of topical consistency for different types of documents.

3 Topical consistency measures

This section discusses two methods of capturing the topical consistency of a text. We start with *lexical cohesion*, a familiar text analysis approach that uses information about the semantic relatedness of words to capture the topical structure of a text. Evidence emerges that the advantages of lexical cohesion are outweighed by its shortcomings. In particular we comment on its lack of sensitivity to topical hierarchy. Thus motivated, we introduce a second method of capturing topical consistency, the *coherence*. We propose a coherence

Table 2 The low-noise blog excerpt (cf. Fig. 1a) generates seven unique strong lexical chains, while the moderate-noise excerpt (cf. Fig. 1b) generates nine

Strong lexical chains the low-noise blog excerpt in Fig. 1a (five posts of "Growing Vegetables")
Garden (plant, sow, bed, seed, gardeners, weed, plants, planted, weeding, landscape, beds, sown, cultivate, seeds, weeds, garden)
Soil (building, yard, side, ground, stone, walk, rows, soil)
Raised (realize, fruit, raised, harvesting, clearing, finding, crops, bring, light, crop, produce)
Space (reach, keeping, wide, spacing, foreign, space, spread)
Easy (leg, sitting, easy, easier, maintain, summer, arm, proper, giving, maintaining)
Grow (time, grow, half, growing)
Deep (huge, sizes, deal, larger, run, deep, size)
Grow (discourage, start, grow, care, growing)
Strong lexical chains in the moderate-noise blog excerpt in Fig. 1b (five posts of "Storybook Cottage and Gardens")
Planted (plant, manure, seed, bed, green, plants, planted, nursery, plot, winter, beds, seeded, hoping, gardening, dig, seeds, garden)
Sprouts (fill, biggest, wide, growing, blew, putting, sprouting, spring, pot, grown, full, grow, sprouts, sprout, develop, pots)
Bit (dish, root, breakfast, crop, dinner, cookie, super, picking, bit, takes, beets, foods, food, lettuce, crops, eating, eat, pick, square)
Row (fit, warm, thunderstorm, ran, fly, heads, weather, sets, heading, row, rows)
Left (post, yellow, double, posted, reference, spot, typical, figure, red, forward, blue, left, notes, note)
Time (time, woke, days, beans, day, fun, tapes, Day)
Starts (starts, start, die, starting, started)
Batch (showers, lots, batch, lot, pack, packs, closely)
Plans (plans, plan, wire, advise, suggest, plain, explains, planned, advice)

Chains are ordered by decreasing chain strength; keywords are shown in bold

score that captures the clustering structure of a document collection. In a series of examples with synthetic data, including text data, our proposed coherence score is shown to have properties desirable for capturing topical noise. The section concludes with a summary of the benefits of the coherence score, including the advantages of choosing the coherence score as a measure of topical consistency.

3.1 Lexical cohesion

The concept of cohesion [15] is used in text analysis to describe the topical relationships between various units of text. Cohesion is a set of characteristics that conspire to make text "stick together" topically [5]. Lexical cohesion measures cohesion by examining the semantic relationships between the content words used in a text [28]. Lexical cohesion is easy to identify [5] and can be calculated automatically using an appropriate linguistic resource such as a thesaurus. Semantically similar words (usually nouns) occurring in close proximity to one another build lexical chains, which indicate that a unit of text is about the same topic [28]. Lexical chains form the basis for models of lexical cohesion [5,28,34]. A primitive form of lexical cohesion does not make use of similar lexical words, but rather measures repetition of the same word form or forms. The *cohesiveness filter* proposed by [2] encodes an entropy-based measure of query-word repetition patterns and, as such, is an example of this primitive form of lexical cohesion. The disappointing results of this filter as applied to the task of identifying topically focused web pages in the TREC-2003 Web Track topic distillation task motivate us to turn our consideration to full-fledged forms of lexical cohesion that look beyond word-form repetition and make use of external resources to derive information concerning lexical similarity.

A priori, lexical cohesion is an appealing approach to capturing topical consistency. It is intuitive that the topical diversity of a text is reflected in the number of distinct topics it discusses. The number of topics in a text, in turn, is reflected by the number of lexical chains of words with similar meanings that the text contains. The low-noise blog excerpt in Fig. 1a and the moderate-noise blog excerpt in Fig. 1b are convenient examples that provide an impression of how lexical chains capture topical consistency. We generate *strong lexical chains* from these blog excerpts using the LexicalChain application of the electronic lexical knowledge base (ELKB),[3] which is based on Roget's Thesaurus and implements the algorithm proposed by Barzilay and Elhadad [5]. The chains are shown in Table 2. The LexicalChain algorithm computes lexical chains by clustering words that are both semantically similar and near to each other in the text. *Chain score* is the length of the chain as measured by the number of words it contains weighted with a factor reflecting the number of repeated words. *Strong chains* are defined as chains that have a score greater than the mean score plus

[3] http://www.nzdl.org/ELKB. Accessed 19 May. 2009.

two standard deviations. The highest frequency member of a chain is defined to be its *keyword*. From Table 2, it can be observed that the low-noise blog excerpt generates eight strong lexical chains, seven of which have a unique keyword. The moderate-noise blog, on the other hand, generates nine strong lexical chains. The difference in chain number reflects human intuitions about the topical diversity of the two blogs. The difference is not strikingly large, but still serves to illustrate the way in which intuitions of topical diversity are related to the number of topics as reflected by the number of lexical chains a text contains. Other five post excerpts of the same blogs display similar differences in chain number.

In addition to providing an impression of how lexical cohesion works, this example also illustrates one of its shortcomings. Lexical cohesion is sensitive to the progression of topics in a text, but is rather blind to their hierarchical structure. Where humans may differentiate between a central and a subordinate topic, the LexicalChain algorithm produces two lexical chains of approximately the same length. For example, in Table 2 it can be seen that in the low-noise topical blog, a chain with the keyword "soil" is produced, which is a plausible central topic of the blog. A chain with the keyword "space" is also produced, which arises due to mention of spatial concepts in various contexts, but is less likely to be understood as an actual topic of the blog. It is challenging to determine the topical consistency of a text collection by using lexical chains to count the number of distinct topics occurring, since it is not readily obvious which chains to count as representing central topics of the text.

The problem of distinguishing central from subordinate topics can be circumvented by setting aside the chain-based lexical cohesion approach, and instead looking directly at the inherent clustering structure of the collection, i.e., the topic groups that emerge when the documents in the collection are compared to each other. In the next section, we introduce a measure that captures the clustering structure of the collection, the coherence score, and demonstrate its desirable properties for capturing topical consistency.

3.2 Coherence score

The 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. The coherence score we propose derives its inspiration from the *expression coherence* score used in the genetics literature [30].

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 documents 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:

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

where the similarity between documents d_i and d_j is a similarity or distance metric[4] describing the semantic closeness of the two documents. Theoretically, it can be any similarity/distance measures, however, according to [35], different similarity/distance measures have different features and selection of the measure would be application dependent. In our experiments, we use the cosine similarity, which is widely used and proven to be effective in our previous experiments with coherence measures.

We follow [18] and define the *coherence* (Co) of the document set D to be

$$Co(D) = \frac{\sum_{i \neq j \in \{1,\ldots,M\}} \delta(d_i, d_j)}{\frac{1}{2}M(M-1)}. \tag{2}$$

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 (i.e., the full corpus of posts) 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 κ fraction as the value of τ'. Heuristically, we set κ 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 actual 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.

As an example, consider the blog examples discussed above; they generate coherence scores consistent with our expectations. The coherence score of the blog excerpt with low topical noise (cf. Fig. 1a) is 0.5 compared to 0.3 for the blog excerpt with moderate topical noise (cf. Fig. 1b). We now proceed with toy examples that will demonstrate more rigorously the power and reliability of the coherence score in measuring the topical structure of a set of documents.

[4] Note that if a distance metric is used, the criterion of a pair of documents being "coherent" is that the distance between the pair should be *lower than* a given threshold.

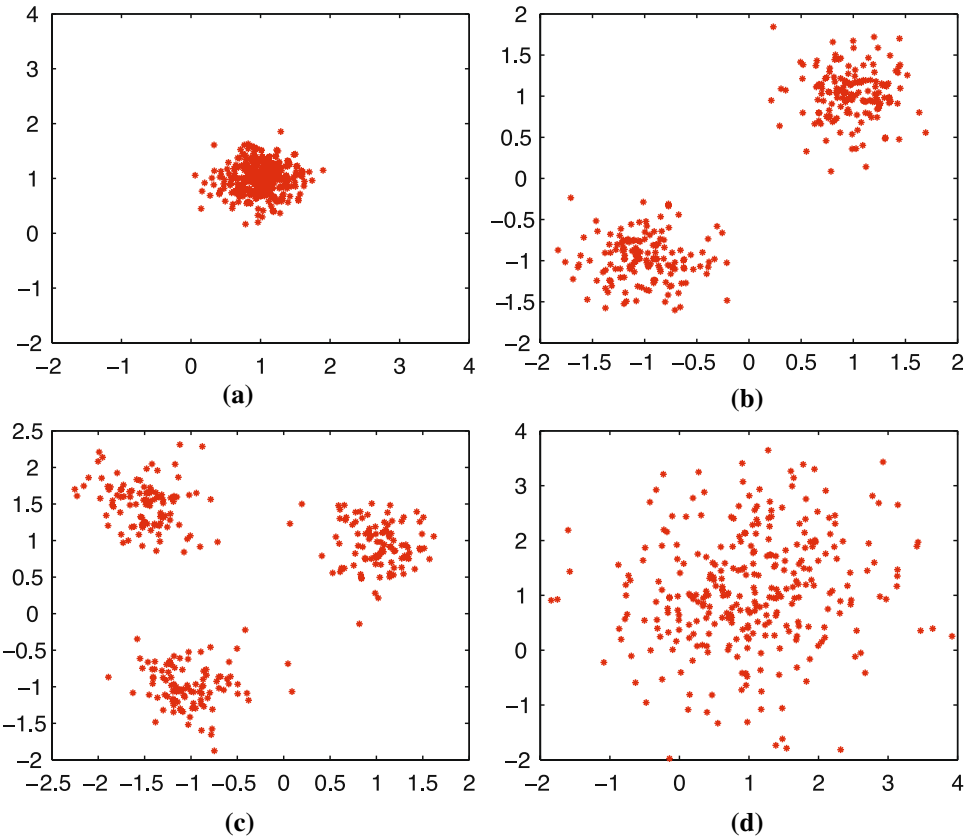

Fig. 2 A toy example. (**a**) 1 random sample from a normal distribution with $\mu = (1, 1), \sigma = 0.3$; (**b**) 2 random samples from a normal distribution with $\mu_1 = (-1, 1), \mu_2 = (1, -1), \sigma_1 = \sigma_2 = 0.5$; (**c**) 3 random samples from a normal distribution with $\mu_1 = (-1, 1),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

3.2.1 Toy example 1: 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 (d) 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 sets (b) and (c) consist of 2 and 3 sub-clusters, respectively. Data set (a) consists of a tight cluster. Figure 2 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 3 shows the results. We can see that, ranking in terms of total variance, we have $(a) > (d) > (b) > (c)$; while in terms of coherence, we have $(a) > (b) > (c) > (d)$, 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.

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 feed search task. When measuring the recurring interest in a given topic X, assuming all blogs have some posts relevant to the topic X, one would expect that the blogs with posts that concentrate on one topic to display stronger recurring interest than blogs that discuss multiple topics.

Table 3 The coherence score and the total variance of the toy data sets

Datasets	(a)	(b)	(c)	(d)
Coherence score	0.0092	0.0056	0.0034	0.0006
Total variance	0.1748	2.1728	2.5315	2.1227

3.2.2 Toy example 2: coherence score on text data

In order to test the power and reliability of the coherence score in measuring the topical structure of a set of documents, we perform experiments using synthetic text data, which is constructed using real world data whose clustering structure is kept strictly under control.

We artificially construct four data sets each containing 60 documents taken from TREC test collections (see Sect. 2.2). The first three data sets are generated by randomly sampling 1, 2, and 3 queries from TREC topics (i.e., TREC queries), and extracting the relevant documents from TREC qrels (i.e., TREC relevance judgment sets). In this way, we control the topical structure of the document set by varying the number

Table 4 The mean value and variance of the coherence scores for clusters containing different number of topics, each setting is sampled 100 times with cluster size of 60 documents

	Scores	1-topic clusters	2-topic clusters	3-topic clusters	Random sets
AP89+88	Mean (var)	0.7205 (0.0351)	0.4773 (0.0199)	0.3900 (0.0284)	0.0557 (0.0002)
Robust04	Mean (var)	0.6947 (0.0463)	0.4490 (0.0114)	0.3365 (0.0064)	0.0457 (0.0002)
Blog06	Mean (var)	0.6663 (0.0378)	0.5215 (0.0226)	0.4405 (0.0126)	0.0495 (0.0003)

The experimental results on the AP89+88 and Robust04 collections are directly taken from our previous work [17]

of topics it contains. The fourth data set is a random set, sampled directly from the background collection. We calculate the coherence score for each data set. The construction procedure for the four data sets is repeated 100 times. Table 4 shows the average coherence score for these 100 runs on different TREC collections (see Sect. 2.2).

The results in Table 4 reveal that on average, the data sets with a 1-topic cluster have obviously higher coherence scores than the data sets with 2- and 3- topic clusters and the random data set. Although the collections are composed of documents of a different nature, i.e., news articles as well as user-generated content (blogs), the behavior of the coherence score is consistent. This experiment promises that the coherence score does indeed reflect the topical structure of a set of documents.

3.3 Advantages of the coherence score

We have seen that the coherence score is able to capture the clustering structure of data, and in particular, the topical consistency of text. The coherence score holds clear potential for capturing the topical consistency of user-generated content. We close this section with a summary of the advantages of using the proposed coherence score as a measure of topical consistency for blogs.

First, the coherence score relies only on the statistics derived from the collection and is independent of any external resources. In order to calculate alternate measures such as lexical cohesion, an external knowledge resource such as a thesaurus or lexical database such as WordNet [13] is necessary. Dependence on external resources raises several issues. The costs of licensing comes immediately to mind. More critical, however, is that external resources often fail to be up-to-date with regard to proper nouns [34], which is especially needed in a fast-changing environment like the blogosphere. Further, we must be able to filter our collection and regard blogs written only in languages covered by available resources, a challenging task in face of the fact that some bloggers switch languages while posting. In these respects, using the coherence score offers clear benefits of independence and flexibility.

Second, the coherence score does not require optimization of parameter settings. For the coherence score, the only parameter is the threshold τ, which defines the "non-randomness" for a given collection. Recall that τ is set by sampling the background collection for a given κ. Although κ is determined heuristically, our previous experiments show that the value 0.05 is proved to be quite stable. Coherence is thus easier to apply than measures such as lexical cohesion. In order to build lexical chains, the setting of two parameters is required: a threshold on the semantic relatedness of two words and a threshold on the physical distance, i.e., the number of words separating them in the running text. These parameters determine whether a word should be added to an existing chain or start a new chain [34]. Presumably, parameter settings would have to be re-optimized for a new corpus.

Third, the coherence score directly captures the clustering structure of the collection. For this reason, it is not necessary to be concerned about identifying individual topics or their relative importance in the blog. As discussed above, a lexical cohesion measure based on lexical chains encounters the challenge of distinguishing chains representing central topics from chains representing subordinate topics. Although we do not exclude the possibility that further development work would allow this issue to be addressed, the coherence score approach offers the advantage of circumventing the issue entirely.

Fourth, the coherence score is relatively efficient to compute. Its computational complexity is $O(s \cdot n^2)$, while the complexity of a typical lexical chain algorithm is $O(s^2 \cdot n^2)$, where s is the average length of the individual documents in words and n is the number of documents in the document set on which the coherence measure is performed. Although in practice the computational complexity of the calculation of lexical chains can be kept well below its theoretical limit, it still fails to be competitive with that of the coherence score. For our experiments, we calculate the coherence score for the set of blog posts in a given blog. The coherence score is calculated offline at indexing time, i.e., we calculate the scores once for all blogs in the collection. With our implementation, the calculation of pairwise cosine similarity scores takes around 1.5 seconds for 500 documents. Table 5 shows the dis-

Table 5 Distribution of the blog lengths, i.e., number of posts contained in a blog

Total number of blogs: 83,320					
Blog length	<10	10–50	50–100	100–500	>500
Number of blogs	21,290	42,085	15,338	4,514	103

tribution of the number of posts in blogs, which provides an impression of the feasibility of our approach.

Finally, as was demonstrated by the toy examples, the application of coherence score is not even limited to text data. Information other than words such as structure of the documents, hyperlinks contained in the web pages, etc., could be easily integrated.

These advantages provide motivation for us to leave aside consideration of measures with the disadvantages of lexical cohesion and continue our investigation by testing the efficacy of the coherence score. In particular, we investigate whether the coherence score can be exploited to model topic consistency and improve retrieval in the blog feed search task.

4 Using coherence in the setting of blog feed search

In this section we detail the modeling of the task we address: modeling topical noise in user-generated content. To this end, we first explain our blog feed search modeling framework in Sect. 4.1; after that we introduce alternative ways of incorporating the coherence score in this framework (Sect. 4.2).

4.1 Blog retrieval model

Our approach to modeling blog feed search, first introduced in [4] is based on expert retrieval models [3]. 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 units, and (iii) the most important reason, to allow the use of one index for both blog post and blog feed search [36].

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 $p(\text{blog}|q)$, and rank blogs according to this probability. Since a query generally consists of only a few terms, often underrepresenting the information need that gave rise to it, Bayes' Theorem is applied in order to achieve a more accurate estimate:

$$p(\text{blog}|q) = \frac{p(q|\text{blog}) \cdot p(\text{blog})}{p(q)}, \tag{3}$$

where $p(\text{blog})$ is the probability of a blog: in our baseline approach $p(\text{blog})$ is assumed to be uniform, that is $p(\text{blog}) = |\text{blog}|^{-1}$, where $|\text{blog}|$ is the number of blogs in the collection; other ways of estimating $p(\text{blog})$ are detailed in Sect. 4.2. The component $p(q)$ indicates the probability of a query. In the remainder of the paper, we refer to the retrieval status value (RSV) rather than to $p(\text{blog}|q)$. This terminological shift is necessary since our experiments involve incorporating scores into $p(\text{blog}|q)$ that have the same scale as probabilities, but are not otherwise true probabilities.

As a common practice in language modeling approaches, $p(q)$ is discarded as it does not affect the ranking of the results (for a given query q). However, when the impact of the coherence score is taken to be a function of the RSV (as we will discuss in Sect. 4.2), 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 [20], who estimate the probability of a term $p(w)$ with following equation:

$$p(w) = \sum_{m \in M} p(w|m)p(m), \tag{4}$$

where w is a term and M is a set of relevance models. We can translate this equation to our blog feed search model by replacing $p(w)$ with $p(q)$ and M with B, a set of blogs. We end up with Eq. 5:

$$p(q) = \sum_{\text{blog} \in B} p(q|\text{blog})p(\text{blog}). \tag{5}$$

We set B to be the top 200 results, i.e., retrieved blogs, for query q so as to estimate $p(q)$.

Next, we focus on the estimation of the query likelihood, $p(q|\text{blog})$: 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 t given the blog model is $p(t|\theta_{\text{blog}})$. 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:

$$p(q|\theta_{\text{blog}}) = \prod_{t \in q} p(t|\theta_{\text{blog}})^{n(t,q)}, \tag{6}$$

where $n(t, q)$ is the number of times term t is present in query q. In order to prevent data sparseness 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 $p(t|\text{blog})$, which is then smoothed with term probabilities $p(t)$ estimated using the background collection:

$$p(t|\theta_{\text{blog}}) = \lambda_{\text{blog}} \cdot p(t|\text{blog}) + (1 - \lambda_{\text{blog}}) \cdot p(t). \tag{7}$$

In Eq. 7, $p(t)$ 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. We discuss the estimation of the smoothing parameter λ_{blog} in Sect. 5.

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

$$p(t|\text{blog}) = \sum_{\text{post} \in \text{blog}} p(t|\text{post}, \text{blog}) \cdot p(\text{post}|\text{blog}), \tag{8}$$

We make the assumption that the post and the blog are conditionally independent, setting $p(t|\text{post}, \text{blog}) = p(t|\text{post})$. The importance of a given post within the blog is expressed by $p(\text{post}|\text{blog})$. 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, $p(\text{post}|\text{blog}) = \text{posts}(\text{blog})^{-1}$, where posts(blog) is the number of posts in the blog.

4.2 Incorporating the coherence score into the blog retrieval model

Now that we have outlined our blog retrieval framework, we shift our attention to the incorporation of the coherence score in this framework. Before we jump to actually modeling this, we take a step back and look at the relation between the coherence of a blog and its relevance regarding a topic. In case of a (topically) relevant blog, this blog should not be highly favored in the final ranking unless it is *also* topically coherent. On the other hand, if we have a blog that has high topical coherence because it consistently treats a different topic than the relevant topic, we do not want this blog to enjoy an unjustified promotion within the final ranking. Instead, we would like to target a more desirable behavior: blogs that are ranked high for a given topic should enjoy a boost from the coherence score that allows them to maintain their prominence while bottom ranked blogs should be prevented from deriving benefit from their coherence score; in the latter case the chance is greater that they are coherent with respect to non-relevant topics. Finally, documents in between should be given a moderate advantage if their coherence scores are high. We can look at this desirable behavior as *local* re-ranking in contrast to *global* re-ranking, which allows for a document to take a brutal jump from the very bottom to the very top of the final ranking.

A transparent, straightforward integration of coherence in our retrieval framework 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 $p(\text{blog})$. As detailed in Sect. 3.2, the coherence score is already a proportion, which means that it is scaled like a probability, and for this reason we can simply estimate

$$p(\text{blog}) = Co(\text{blog}) \tag{9}$$

where $Co(\text{blog})$ is calculated using Eq. 2, and the threshold τ is estimated to be 0.1, given that the κ is set to 0.05 heuristically. 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 (0.01). 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 recurring interest in a topic.

Although the implementation of coherence as a prior is straightforward, it does not fulfill the properties we discussed in the first paragraph of this section: 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. One issue here is that we do not have relevance judgements for our ranked documents. Instead, we use the baseline retrieval score RSV of a blog with a uniform prior (viz. Eq. 3), as a substitute for judged relevance. We prefer the retrieval score of the blog over an obvious alternative, using the rank of the blog in the retrieval result list. If the rank were used, a small difference in RSV could have a disproportionately large impact on the rank, making the weights over-sensitive 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 the RSV scores. In particular, we want blogs with RSVs close to 0 to receive nearly no contribution from the coherence score while the blogs with the 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*)

$$W(\text{RSV}) = \text{RSV} \tag{10}$$

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

$$W(\text{RSV}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\text{RSV}-\mu)^2}{2\sigma^2}\right) \tag{11}$$

Quadratic function 1 (*quad1*):

$$W(\text{RSV}) = \text{RSV}^2 \tag{12}$$

Quadratic function 2 (*quad2*):

$$W(\text{RSV}) = -(\text{RSV}-1)^2 + 1 \tag{13}$$

Mixed function of 12 and 13 (*qmix*) with γ as the free parameter:

$$W(\text{RSV}) = \begin{cases} \text{RSV}^2 & \text{if RSV} < \gamma, \\ -(\text{RSV}-1)^2 + 1 & \text{otherwise.} \end{cases} \tag{14}$$

This choice of functions allows us to explore a linear relation (Eq. 10), a non-linear relation with different rates of increase (Eqs. 11, 12, 13), and a combination of different rates of increase (Eq. 14). Figure 3 shows the curves of these functions in order to provide an intuition of the properties of the functions.

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

$$wCo(\text{blog}, \text{query}) = W(\text{RSV}) \cdot Co(\text{blog}) \tag{15}$$

The experimental models use wCo as the blog "prior," substituting for $p(\text{blog})$ in Eq. 3, leaving us with the final ranking equation

$$\text{RSV} = \frac{p(q|\text{blog}) \cdot wCo(\text{blog}, \text{query})}{p(q)}. \tag{16}$$

In summary, from our observations regarding the relation between coherence and relevance, we introduce two main methods for incorporating the coherence score into our retrieval framework: (i) a query-independent method, using $Co(\text{blog})$ directly as $p(\text{blog})$, and (ii) a relevance-dependent method, where $Co(\text{blog})$ is weighted using a function of the RSV. The latter method is translated into five weighting functions. In the next section we detail our experimetal setup, before comparing results of the different implementations in Sect. 6.

5 Experimental setup

For our experiments on blog feed search we use the blog collection introduced in Sect. 2.2. The TREC 2007 Blog track supplies 45 blog feed search topics, also referred to here as queries, and assessments concerning which blogs are relevant to which topics [24]. 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).

5.1 Metrics and significance

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).

We determine statistical significance of differences using a Wilcoxon paired signed rank test with $\alpha = 0.05$.[5] Significant changes are indicated using ▲ (significant increase) or ▼ (significant decrease).

5.2 Smoothing

The performance of language modeling-based retrieval methods is highly responsive to smoothing [40]. To estimate the smoothing parameter λ_{blog} in Eq. 7 in our model, we set λ_{blog} equal to $\frac{n(\text{blog})}{\beta + n(\text{blog})}$, where $n(\text{blog})$ 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 [25]). 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, $\beta = 17{,}400$).

[5] We also perform the one-tailed t test which leads to a similar conclusion as the Wilcoxon test.

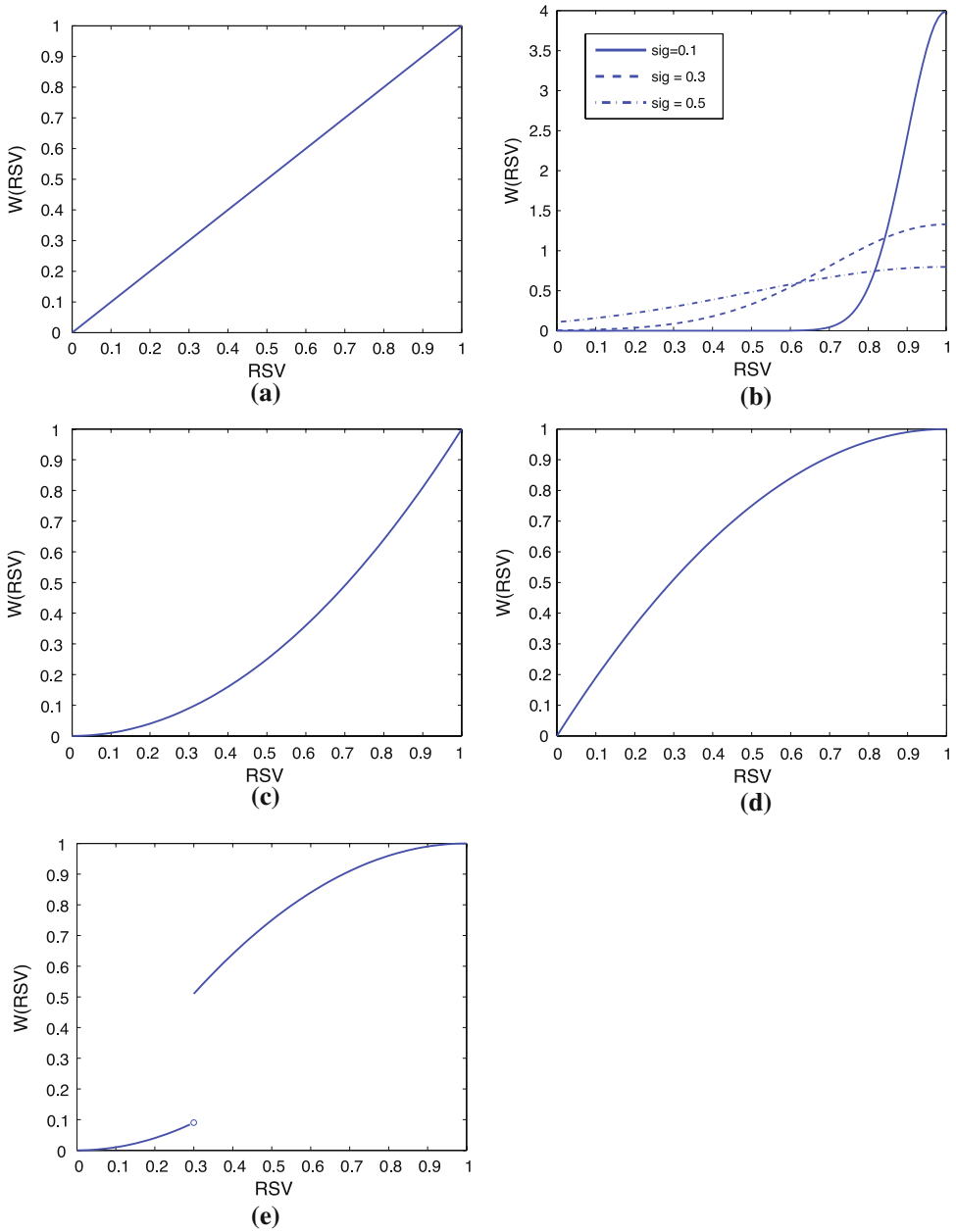

Fig. 3 Weighting functions. **a** Linear (lin). **b** Normal distribution (*norm*). **c** Quadratic 1 (*quad1*). **d** Quadratic 2 (*quad2*). **e** Quadratic mixed (*qmix*) example where $\gamma = 0.3$

5.3 Parameter estimation

For functions *norm* and *qmix* we need to set parameters σ and γ. We performed a sweep over possible (and sensible) values of both parameters ($0 < \sigma < 1.0; 0 < \gamma < 0.1$) and evaluated the performance on MAP. Based on the results of the sweep, we select $\sigma = 0.05$ for *norm* and $\gamma = 0.05$ for *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.

6 Results

Let us revisit the research questions introduced in Sect. 1: Our first question, *what is a proper way of estimating the coherence of a blog?*, has already been addressed in Sect. 3 where we offer our coherence score as a solution.

In this section, we turn to the second question, *how can we use the coherence score in our retrieval process?* A number of options, ranging from treating the coherence as a simple prior to modeling it as a multiplicative factor whose contribution is a function of the RSV of a blog, are proposed in Sect. 4. We now compare the results of these options, analyze

the outcomes, and compare our results to other systems. In Sect. 7, we look into the third research question of *how the sample size influences estimating coherence and what the impact is on blog feed search*.

6.1 Baseline versus coherence-based results

Table 6 lists the results for our baseline model, *baseline*, which uses a uniform prior, our straightforward implementation of coherence, *prior*, which uses Co(blog) (cf. Eq. 2) as prior, and the five experimental models, designated *lin*, *norm*, *quad1*, *quad2*, and *qmix* according to which version of the weighted coherence score wCo they integrate.

The run using coherence as a prior performs significantly worse than the baseline in terms of MAP, but shows slight (non-significant) improvements on early precision (P@5) and MRR. We can see that all weighting functions show some improvement over the baseline, with the function *qmix* performing best in terms of MAP and MRR. 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 and MRR, but also shows improvements on P@5, P@10 in most cases, although not significant.

Let us take a closer look at the results per topic (i.e., query). In Fig. 4 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) the 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 the maximum increase in AP is high (0.15 for topic 974), but so is the maximum drop (−0.14 for topic 979), (v) the function *quad2* improves performance in 34 of 45 topics, but also shows a large drop for several topics, and finally (vi) the function *qmix* improves over the baseline in 35 of 45 topics, with a limited drop in AP for the worst performing topic (−0.07 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 (i) related to the breadth of the vocabulary that a blogger uses to discuss the topic, (ii) the ability of the topic to inspire bloggers over time and (iii) spam blogs whose word distributions cause them to be relevant to that topic.

What happens when we explore the per-topic differences between the run using coherence as a prior and the runs using the weighting functions? Overall trends show three topics score worse using weighting functions compared to the prior: 953 (*biofuels may damage forests*), 957 (*Russia*), and 992 (*copyright law*). On the other hand we see three topics that are in the top 3 of most improved topics over the prior run (for all weighting functions): 974 (*tennis*), 973 (*autism*), and 954 (*Mac*). In general, very few topics actually perform better in the prior run than using the weighting functions (8–12 topics out of 45).

Finally, we look at the differences between the runs using the various ways of weighting coherence and see what causes the final evaluation results to be different: Are certain topics hurt by one function, but improved by another? Or do we see a general trend of topics improving or dropping for all functions, just differing in the degree of gain or loss? We try to answer this question using several topics as examples. First, topics 979 and 982 drop most and second most for all functions. At the other end of the spectrum, we have similar, consistent behavior for topics 974 (improves most), 994 (improves second most), and 995 (improves third most). Only few topics show different behavior: topic 964 (*violence in sudan*) improves for *norm*, *qmix*, and *quad2*, but drops for *lin* and *quad1*. Also, topic 992 (*copyright law*) drops in all cases, except for *norm*. The overall picture however, shows consistent behavior for topics over all functions, with the level of improvement (or loss) making up for the differences in MAP between the runs.

Table 6 Results of coherence score, implemented as prior, and 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*)

Function	MAP	P@5	P@10	MRR
Baseline	0.3272	0.4844	0.4844	0.6892
prior	0.2945▼	0.5022	0.4822	0.6959
lin	0.3326	0.5022	0.**5067**	0.7266
norm	0.3325▲	0.5022	0.4822	0.7103
quad1	0.3327	0.5022	0.**5067**	0.7377
quad2	0.3365▲	0.5022	0.5022	0.7154
qmix	**0.3382**▲	0.**5067**	0.5022	**0.7394**▲

Significance computed against the baseline

6.2 Comparison to other systems

In this section, we compare our results to the performance of other systems on the blog retrieval task. One of the baselines is the system described in [23]. In Sect. 2.1, we had a brief discussion of the approaches for modeling central or recurring interest introduced by [23]. On top of that, the paper also tries to incorporate different kinds of heuristics to improve retrieval performance, such as using different document weighting models, using term proximity features, blog

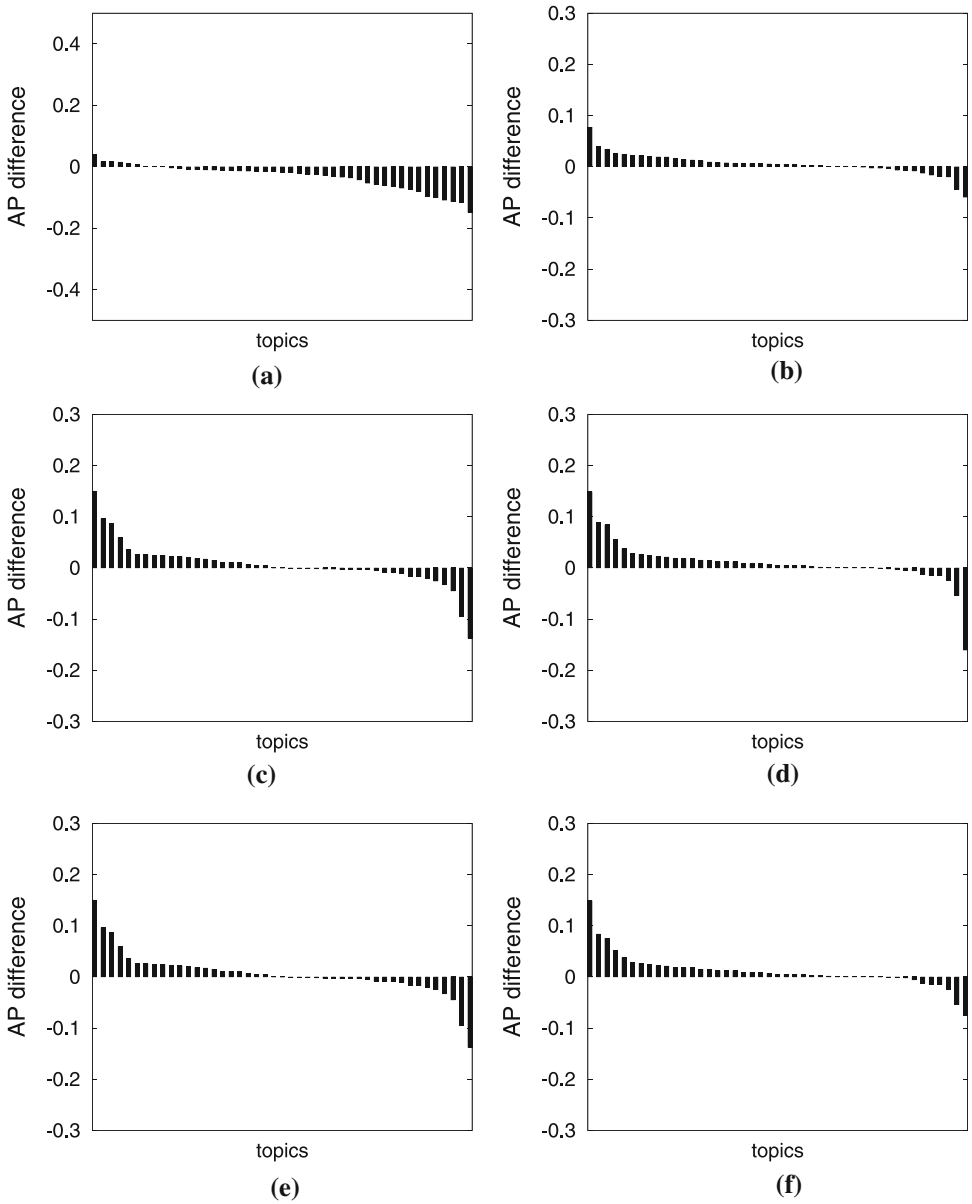

Fig. 4 AP differences between baseline and (*left*-to-*right*, *top*-to-*bottom*) coherence as prior, *norm*, *quad1*, *quad2*, *lin*, and *qmix*. **a** prior. **b** norm. **c** quad1. **d** quad2. **e** lin. **f** qmix

size normalization and enrichment (i.e., query expansion on Wikipedia).

In Table 7, we list some of the results reported by [23], which is referred to as *keyBlog*. Specifically, we compare our performance to the keyBlog system at three levels: (i) the results using recurring interest (Date feature), which is shown to be the most successful feature for modeling the topical concentration or recurring interest in the keyBlog system; (ii) the best results in the keyBlog system without enrichment, since our system does not involve any query expansion. The system includes the following elements: field based document weighting model (PL2F), blog size normalization, Date feature and proximity features; (iii) the overall best results of the keyBlog system, which is similar to (ii), but includes external query expansion in Wikipedia.

We also include the TREC best run as reference scores. Since the TREC best run uses external query expansion as well, we separate the table into two parts: with and without external query expansion. For details of the system design for the TREC best run, we refer to [10].

In terms of MAP without external query expansion, our method achieves a better performance than the keyBlog system. Besides the end-to-end performance comparison, we argue that our method is more principled and efficient in that we do not require large amounts of parameter-tuning nor do we require efforts on query expansion with additional corpora, which were employed by both other systems. On the other hand, the external query expansion is an interesting approach that greatly improves the performance of both systems that we are comparing to, cf. also [37]. We leave it as

Table 7 Comparing our methods to the best run reported in [23] (keyBlog) and TREC best run. Best score of each category and each metric uses boldface

	MAP	P@10	MRR
Systems without external QE			
keyBlog with Date	0.2788	0.5022	0.**7893**
keyBlog best without Enrichment	0.3187	0.**5800**	0.7798
qmix	0.**3382**	0.5067	0.7394
Systems with external QE			
keyBlog best	0.3481	0.**6044**	0.**8405**
TREC best run	0.**3695**	0.5356	0.7537

future work to examine the interplay between coherence and external expansion.

7 Impact of sample size on coherence and blog feed search

Following the assumption we made in our blog retrieval model, for each blog there is a blog model that generates the texts we observe. Since the Blog06 collection is crawled in a certain period, for a given blog we can see it as a sample drawn from an underlying distribution generated according to the blog model. One would expect that the judged relevant blogs, i.e., blogs having recurring interest on a given topic, are generated by blog models that generate blog posts with topical consistency. However, since we only see the posts collected during 11 weeks, the true topical distribution of the blog is actually approximated by this observable sample.

Intuitively, in order to get a good estimation of the coherence of the underlying topical structure of the blog model, a certain number of posts should be contained in the sample under observation. This leads us to the following questions. What is the impact of sample size on the estimation of the coherence of the true topical structure of the underlying blog model? Can we decide on a minimum number of posts to achieve a reliable estimation? And how would this threshold impact blog feed search performance? Below, we address these questions with exploratory experiments.

7.1 Impact of sample size on the estimation of coherence

Intuitively, we expect that a larger sample will have a better approximation of the true topic distribution of the population, i.e., a blog with more posts within the 11 week period of the data set should have a better approximation of the distribution of the topical structure of the blog in an infinite amount of time. Moreover, it is also intuitive that populations of different sizes require different minimum sample sizes for a reliable approximation. Since we do not know the size of the population, i.e., we do not know the number of posts a blog contains outside the 11 weeks covered by the data set, we need to decide on a minimum number of posts that would be sufficient for populations of different sizes.

To this end, we collect blogs with different numbers of posts from the Blog06 collection: blogs with 50 posts, 100 posts, 399–499 posts, 500–999 posts. For each number of these four groups, we sample 50 blogs for experiments.

For each of the blogs B we collected, we calculate its coherence, which we denote as $co(B)$. We then sample a different number of posts: 5, 10, 20, 30, 40, and 50 (for the set of blogs of 50 posts, we ignore the 50 sample posts case), and calculate the coherence score for each sample, which we denote as $co(S^k)$, where $k = 5, 10, 20, 30, 40, 50$ is the sample size. We analyze how the value of $co(S^k)$ approximates the value of $co(B)$ as k changes by calculating the mean squared error (MSE) of the sample coherence scores from the real coherence scores derived from the original blog using Eq. 17. For each sample size, we generate 30 runs.

$$\text{MSE}(co(S^k)) = \frac{1}{n}\sum_i \Big(co(s_i^k) - co(B)\Big)^2, \tag{17}$$

where $i = 1, \ldots, 30$, $S^k = \{s_i^k\}_{i=1}^{30}$ is the set of samples from the 30 runs, which are drawn from the original blog B.

To summarize the trends of the impact of sample size on estimating the real coherence for a blog, we take the average MSE of the 50 blogs of different number of posts. Figure 5 shows the results. We see that as the sample size increases, the average MSE decreases. More importantly, as we see in the plot, after 20 posts, the change of average MSE tends to be stable. This trend applies to blogs with different numbers of posts, which suggests that no matter how large the actual

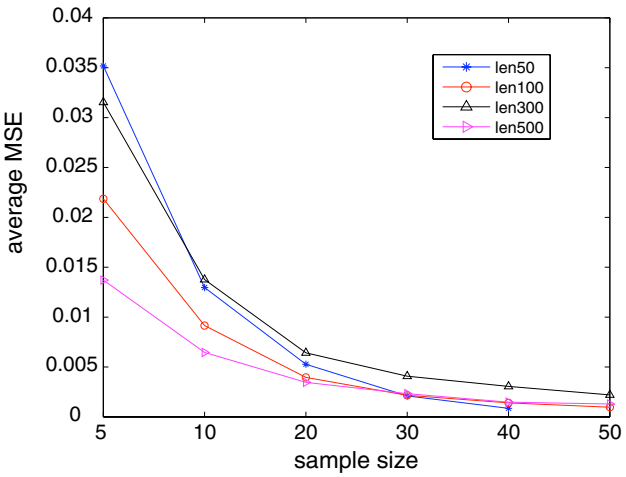

Fig. 5 Relation between the sample size and the average MSE of the sampled coherence score from the real coherence score. Here, *len50*, *len100*, *len300*, *len500* denote the samples of blogs with 50, 100, 399–499, 500–999 posts, respectively

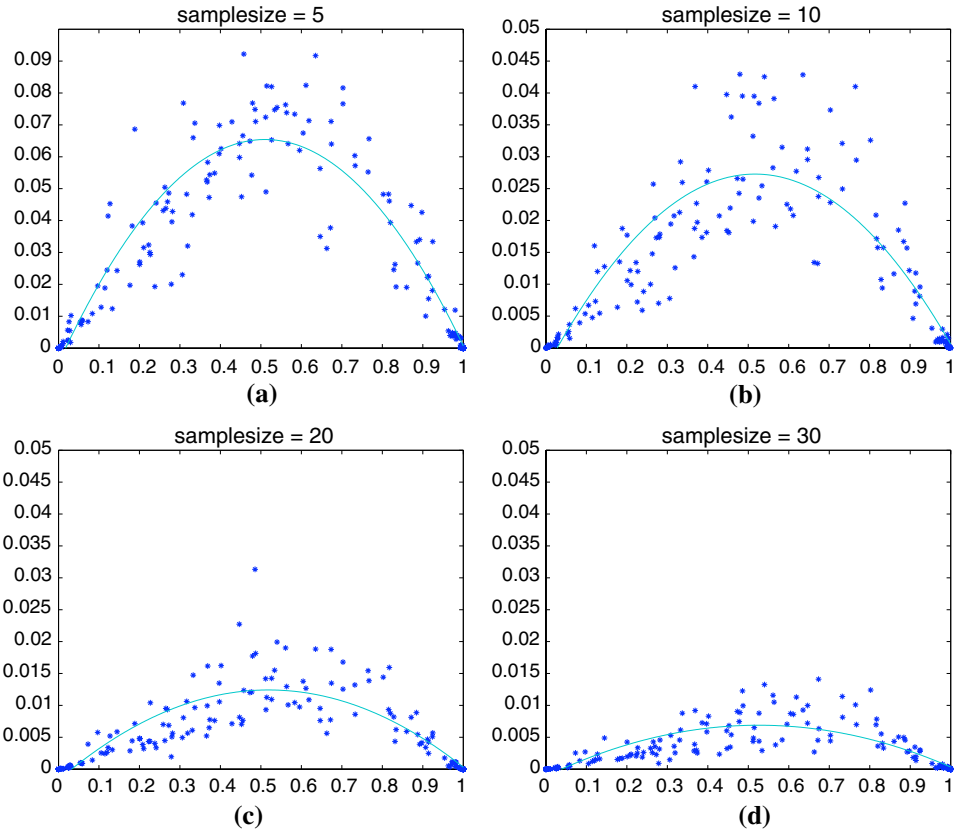

Fig. 6 Relation between the population coherence score (*x*-axis) and the MSE of the sampled coherence score (*y*-axis)

size of the blog would be in an infinite amount of time, a minimum number of 20 posts can achieve a stable estimation of the true topical structure of a blog.

7.2 Relation between the population coherence and the accuracy of being approximated by sampled coherence

One may notice that in Fig. 5, for the same sample size, e.g., 5 posts sample, blogs with 500–999 posts have a lower MSE than blogs with 50 posts. This is counterintuitive. Indeed, we would expect that it is more difficult to approximate the distribution of a large population than a small one with the same amount of samples. In other words, we expect the average MSE of blogs with more than 500 posts to be higher than that of blogs with 50 posts. This unexpected phenomenon suggests that there are other factors besides the sample size that impact the estimation of the topical structure of the underlying blog model. A potential dimension is the coherence of the original blog, i.e., the coherence score of the population.

In Fig. 6, we fix the sample size, and show the relation between the MSE of the sampled coherence score and the population coherence score. We see that the relation is non-linear, but there exists a pattern, which can be approximated by a quadratic function (shown in the plots). Particularly, if the population is extremely coherent, or extremely random, it has a better approximation.

In Table 8, we list the average coherence score of blogs with different numbers of posts that we used in the experiment discussed in Sect. 7.1. As we can see, the average population coherence score of blogs with more than 500 posts is much higher than that of blogs with 50 posts. This explains the phenomenon shown in Fig. 5.

To wrap-up, the experiment in this section shows that for a given post sample size, the coherence of the population is a factor that has impact on the accuracy of the approximation. Populations with extremely random or extremely coherent topical structures are easier to be approximated. The relation between the population coherence and the accuracy of being estimated by sampled coherence is non-linear but has a pattern (i.e., close to a quadratic relation).

7.3 Impact of sample size on blog feed search

Exploring the impact of sampling size a step further, we experiment with post thresholds in the retrieval process. Blogs with fewer posts than the threshold are discarded from the results (both in the baseline setting, as well as in the coherence-based runs), leaving us with a thresholded blog feed search run. We use thresholds between 0 and 50 posts, and use the best performing parameter settings for the five models (i.e., $\sigma = 0.05$ for *norm* and $\gamma = 0.05$ for *qmix*).

Table 8 The average coherence score of the blogs with different number of posts

Blogs of different sizes	50	100	300–499	500–999
Population coherence score	0.5344	0.5755	0.5091	0.7339

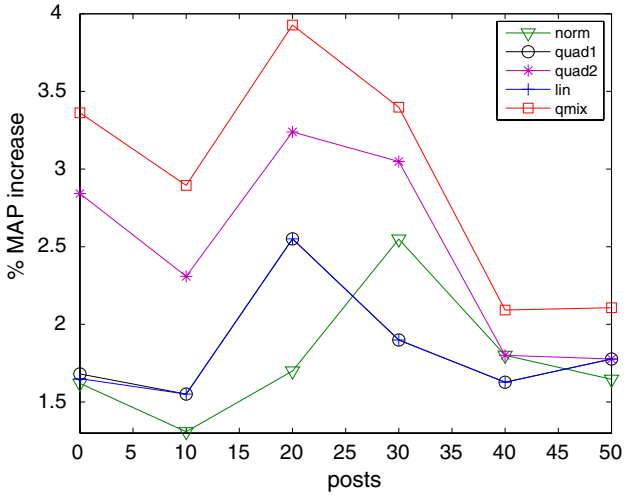

Fig. 7 Effect of threshold on difference in MAP between models and baseline

Figure 7 plots the relative increase in MAP for each of the models over the baseline for different thresholds.

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. On the other hand, if the threshold eliminates too many blogs, the relative improvement will decrease since there may be very few relevant blogs left after thresholding. Table 9 lists the results for each of the functions and the baseline when using a threshold of 20 posts.

The results show that in three cases the improvement over the baseline is significant (in terms of MAP), and that, again, weighting function *qmix* performs best on all metrics.

The experiments in Sects. 7.1 and 7.3 lead to the conclusion that coherence becomes beneficial for blogs when a blog contains more than 20 posts. This result suggests that it would be worth looking into the development of methods to estimate priors for blogs that are (currently) too short to derive benefit from the coherence score.

8 Conclusion

In this paper we proposed a method to counteract the effects of topical noise in blogs with the goal of performing blog feed search. For a blog to be relevant in a feed search 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.

We argued that established cohesion measures, in particular lexical cohesion calculated on the basis of lexical chains, are not suited for measuring topical consistency in the blogosphere and introduced a coherence score which captures the topical clustering structure of a set of documents as compared to a background collection. The coherence score can be calculated relatively efficiently. The calculation makes use of collection statistics only and requires neither external resources nor collection-specific parameter optimization. Applied to blogs, the coherence score reflects topical consistency, in other words, the level of topical noise of a blog.

Incorporating the coherence score in our retrieval framework required us to look at the relation between coherence and relevance: In case of a (topically) relevant blog, this blog should not be highly favored in the final ranking unless it is also topically coherent. On the other hand, blogs that have high topical coherence because they consistently treat a different topic than the given topic, should not enjoy unjustified promotion within the final ranking. To prevent this, we proposed weighting the coherence score by a notion of topical relevance. We compared two methods of incorporating the coherence: (i) a query-independent method, using coherence as prior, and (ii) a relevance-dependent method, where the coherence is weighted using a function of the retrieval score. Results show that the second method outperforms the baseline model, while the first method does not. Furthermore, the *qmix* function performs best with significant improvement over the baseline on MAP and MRR and non-significant improvements on the other metrics.

Following the intuition that the posts in our data set are a sample of the blogger's posts, we expected a larger sample size to be a better approximation of the true distribution of posts. Our analysis of the relation between the sample size and the average deviation of the sampled coherence from the actual coherence of a blog shows that from 20 posts onwards this deviation does not change much anymore, indicating that 20 posts is the minimum sample size needed to get a proper estimation. This is further supported by blog feed search experiments using only blogs that have more posts than a given threshold: using a threshold of 20 posts shows maximum relative improvement over the baseline.

We have shown the coherence score to be effective in capturing topical consistency in user-generated content. Future work will focus on further optimization of the coherence score for use in blog feed search, involving, for example, in-depth investigation of query-specific performance that could lead to further refinement of the weighting function. An extension of the coherence score to other areas of user-

Table 9 Results of weighted coherence score applied to blogs with a minimum of 20 posts. Significance computed against the baseline

Function	MAP	P@5	P@10	MRR
Baseline	0.2470	0.4578	0.4511	0.6930
lin	0.2533	0.4756	0.4689	0.7174
norm	0.2512▲	0.4756	0.4622	0.7030
quad1	0.2534	0.4756	0.4689	0.7285
quad2	0.2550▲	0.4756	0.**4711**	0.7061
qmix	**0.2567**▲	0.**4800**	0.**4711**	0.**7321**

Note that, compared to Table 6, the baseline has changed, due to the fact that blogs with fewer than 20 posts are eliminated from the collection

generated content, such as user reviews or audio blogs (podcasts) is a further avenue of future research.

Acknowledgments We are very grateful to our anonymous reviewers for providing helpful feedback and suggestions. This research was supported by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments (http://www.stevin-tst.org) under project number STE-09-12, and by the Netherlands Organisation for Scientific Research (NWO) under project numbers 017.001.190, 640.001.501, 640.002.501, 612.066.512, 612.061.814, 612.061.815, 640.004.802.

References

1. Allen, R.B., Obry, P., Littman, M.: An interface for navigating clustered document sets returned by queries. In: COCS'93, pp. 166–171. ACM, New York (1993)
2. Amitay, E., Carmel, D., Darlow, A., Herscovici, M., Lempel, R., Soffer, A., Kraft, R., Zien, J.: Juru at trec 2003—topic distillation using query-sensitive tuning and cohesiveness filtering. In: TREC'03 Working Notes (2003)
3. Balog, K., Azzopardi, L., de Rijke, M.: Formal models for expert finding in enterprise corpora. In: SIGIR'06, pp. 43–50. ACM Press, New York (2006)
4. Balog, K., de Rijke, M., Weerkamp, W.: Bloggers as experts. In: SIGIR'08, pp. 753–754. ACM (2008)
5. Barzilay, R., Elhadad, M.: Using lexical chains for text summarization. In: Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL (1997)
6. Bock, H.-H.: On some significance tests in cluster analysis. J. Classif. **2**(1), 77–108 (1985)
7. Cronen-Townsend, S., Croft, W.B.: Quantifying query ambiguity. In: HLT'02, pp. 94–98. (2002)
8. Cronen-Townsend, S., Zhou, Y., Croft, W.B.: Predicting query performance. In: SIGIR'02, pp. 299–306. ACM (2002)
9. Cutting, D.R., Karger, D.R., Pedersen, J.O., Tukey, J.W.: Scatter/gather: a cluster-based approach to browsing large document collections. In: SIGIR '92, pp. 318–329, ACM, NY (1992)
10. Elsas, J., Arguello, J., Callan, J., Carbonell, J.: Retrieval and feedback models for blog distillation. In: TREC'07 Working Notes (2007)
11. Ernsting, B.J., Weerkamp, W., de Rijke, M.: The University of Amsterdam at the TREC 2007 Blog Track. In: TREC'07 Working Notes (2007)
12. Everitt, B.S.: Unresolved problems in cluster analysis. Biometrics **35**(1), 169–181 (1979)
13. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database (Language, Speech, and Communication). MIT Press, Cambridge (1998)
14. Fujimura, K., Toda, H., Inoue, T., Hiroshima, N., Kataoka, R., Sugizaki, M.: Blogranger—a multi-faceted blog search engine. In: WWW'06 (2006)
15. Halliday, M.A.K., Hasan, R.: Cohesion in English (English Language). Longman, London (1976)
16. Hartigan, J.A.: Statistical theory in clustering. J. Classif. **2**(1), 63–76 (1985)
17. He, J., Larson, M., de Rijke, M.: On the topical structure of the relevance feedback set. In: FGIR Workshop Information Retrieval (WIR 2008), Wurzburg, Germany (2008a)
18. He, J., Larson, M., de Rijke, M.: Using coherence-based measures to predict query difficulty. In: ECIR'08, pp. 689–694 (2008b)
19. He, J., Weerkamp, W., Larson, M., de Rijke, M.: Blogger, stick to your story: modeling topical noise in blogs with coherence measures. In: AND '08: Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data, pp. 39–46. ACM (2008c)
20. Lavrenko, V., Croft, W.B.: Relevance-based language models. In: SIGIR '01, pp. 120–127, ACM Press, New York (2001)
21. Lee, W.-L., Lommatzsch, A.: Feed distillation using adaboost and topic maps. In: TREC '07 Working Notes (2007)
22. Macdonald, C., Ounis, I.: The TREC Blogs06 collection: creating and analyzing a blog test collection. Technical Report TR-2006-224, Department of Computer Science, University of Glasgow (2006)
23. Macdonald, C., Ounis, I.: Key blog distillation: ranking aggregates. In: CIKM '08: Proceeding of the 17th ACM Conference on Information and Knowledge Management, pp. 1043–1052, ACM, New York (2008)
24. Macdonald, C., Ounis, I., Soboroff, I.: Overview of the TREC 2007 Blog Track. In: TREC '07 Working Notes, pp. 31–43 (2007)
25. Mackay, D.J.C., Peto, L.: A hierarchical dirichlet language model. Nat. Lang. Eng. **1**(3), 1–19 (1994)
26. Mishne, G.: Applied text analytics for blogs. PhD thesis, University of Amsterdam (2007)
27. Mishne, G., de Rijke, M.: A study of blog search. In: Lalmas, M., MacFarlane, A., Rüger, S., Tombros, A., Tsikrika, T., Yavlinsky, A. (eds.) ECIR'06, vol. 3936, pp. 289–301 (2006)
28. Morris, J., Hirst, G.: Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Comput. Linguist. **17**(1), 21–48 (1991)
29. Ounis, I., Macdonald, C., de Rijke, M., Mishne, G., Soboroff, I.: Overview of the TREC 2006 Blog Track. In: TREC '06 Working Notes. NIST (2007)
30. Pilpel, Y., Sudarsanam, P., Church, G.M.: Identifying regulatory networks by combinatiorial analysis of promoter elements. Nat. Genet. **29**, 153–159 (2001)

31. Robertson, S., Callan, J.: Routing and filtering. In: TREC '05, pp. 99–122. MIT (2005)
32. Seki, K., Kino, Y., Sato, S.: TREC 2007 Blog Track experiments at Kobe University. In: TREC '07 Working Notes (2007)
33. Seo, J., Croft, W.B.: UMass at TREC 2007 Blog distillation task. In: TREC '07 Working Notes (2007)
34. Stokes, N., Newman, E., Carthy, J., Smeaton, A.F.: Broadcast news gisting using lexical cohesion analysis. In: the Proceedings of the 26th BCS-IRSG European Conference on Information Retrieval (ECIR-04), pp. 209–222. Springer (2004)
35. Weeds, J., Weir, D., Mccarthy, D.: Characterising measures of lexical distributional similarity. In: Proceedings of CoLing 2004, pp. 1015–1021 (2004)
36. Weerkamp, W., Balog, K., de Rijke, M.: Finding key bloggers, one post at a time. In: ECAI'08 (2008)
37. Weerkamp, W., Balog, K., de Rijke, M.: A generative blog post retrieval model that uses query expansion based on external collections. In: Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-ICNLP 2009) (2009)
38. Weerkamp, W., de Rijke, M.: Credibility improves topical blog post retrieval. In: Proceedings of ACL-08: HLT, pp. 923–931. Association for Computational Linguisticsages, Columbus (2008)
39. Willett, P.: Recent trends in hierarchic document clustering: a critical review. Inf. Process Manag. **24**(5), 577–597 (1988)
40. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. **22**(2), 179–214 (2004)