Key Blog Distillation: Ranking Aggregates

Craig Macdonald
University of Glasgow
Glasgow, Scotland, UK
craigm@dcs.gla.ac.uk

Iadh Ounis
University of Glasgow
Glasgow, Scotland, UK
ounis@dcs.gla.ac.uk

ABSTRACT

Searchers on the blogosphere often have a need to identify other key bloggers with similar interests to their own. However, a main difference of this blog distillation task from normal adhoc or Web document retrieval is that each blog can be seen as an aggregate of its constituent posts. On the other hand, we show that the task is similar to the expert search task, where a person's expertise is derived from the aggregate of their publications or emails. In this paper, we investigate several aspects of blog retrieval: Firstly, we experiment whether a blog should be represented as a whole unit, or as by considering each of its posts as indicators of its relevance, showing that expert search techniques can be adapted for blog search; Secondly, we examine whether indexing only the XML feed provided by each blog (and which is often incomplete) is sufficient, or whether the full-text of each blog post should be downloaded; Lastly, we use approaches to detect the central or recurring interests of each blog to increase the retrieval effectiveness of the system. Using the TREC 2007 Blog dataset, the results show that our proposed expert search paradigm is indeed useful in identifying key bloggers, achieving high retrieval effectiveness.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; H.3.4 [Systems and software]: User profiles and alert services

General Terms: Performance, Experimentation

Keywords: Blog Distillation, Feed Search, Expert Search

1. INTRODUCTION

The act of blogging has emerged as one of the popular outcomes of the "Web 2.0" phase, where users are empowered to create their own Web content. In particular a (web)blog is a website where entries are commonly displayed in reverse chronological order. Many blogs provide various opinions and perspectives on real-life or Internet events, while other blogs cover more personal aspects. The 'blogosphere' is the collection of all blogs on the Web.

CIKM'08, October 26–30, 2008, Napa Valley, California, USA.

There are several specialised search engines covering the blogosphere, while most of the main search engine commercial players have a blog search product. In their study of user queries submitted to a blog search engine [20], Mishne and de Rijke note two forms of predominant queries: *Context Queries*, and *Concept Queries*. In context queries, users typically appear to be looking at how entities are thought of or represented in the blogosphere - in this case, the users are looking to identify opinions about the entity. In concept queries, the searcher attempts to locate blogs or posts, which deal with one of the searcher's interest areas - such queries are typically high-level concepts, and their frequency did not vary in response to real-world events. For example, for such queries, users are looking for blogs that interest them, so that they can subscribe to these blogs with their RSS reader. The blog search engine should suggest blogs that have posts mostly dedicated to the general topic area of the query - the objective being to provide the user with a list of key (or 'distilled') blogs relevant to the query topic area. For example, a user interested in Formula 1 motorsports would wish to identify blogs giving news, comments and perhaps gossip about races, drivers or teams. Indeed, many of the blog search engines such as Technorati and Bloglines provide a blog search facility in addition to their blog post search facility, while Google Blog Search integrates both post and blog results in one interface. Moreover, many manually-categorised blog directories exist, such as Blogflux and Topblogarea to name but a few. This is reminiscent of the prevalence of the early Web directories (c.f. Yahoo!) before Web search matured, and suggests that there is indeed an underlying user task that needs to be researched [10]. This task is called *blog distillation* [19]. For example, in response to a query, a blog search engine should return blogs that could be added to a directory, or returned to a user as a suggested subscription for his/her RSS reader. Indeed, many blog search engines include the feed URL in their results listing, while Google's RSS Reader provides an integrated feed search application, to allow users to easily find new blogs of interest.

Note that a topic distillation task was developed in the context of the TREC Web Track [4]. In topic distillation, site relevance was required as (i) being principally devoted to the topic, (ii) providing credible information on the topic, and (iii) is not part of a larger site also principally devoted to the topic. Blog distillation is somehow a similar task - the idea is to provide the users with the key blogs about a given topic. However point (iii) from the topic distillation task is not applicable in a blog setting [19].

In general, each blog has an (HTML) homepage, which presents a few recent posts to the user when they visit the

blog. Next, there are associated (HTML) pages known as permalinks, which contain a given posting and any comments by visitors. Finally, a key feature of blogs is that with each blog is associated an XML *feed*, which is a machine-readable description of the recent blog posts, with the title, a summary of the post and the URL of the permalink page. The feed is automatically updated by the blogging software whenever new posts are added to the blog.

Due to this common structure of each blog, a central difference of the blog distillation task from classical Web document search is that a blog can be interpreted as an aggregate of its constituent blog posts, and hence when searching for key blogs, each relevant blog post can be considered as evidence that its corresponding blog is relevant to the query. A natural question is how the blog post-level evidence of relevance should be represented and combined. In this work, we examine two approaches, namely combining all post evidence for one blog into a large virtual document before scoring in response to a query, or combining blog evidence of relevance after the post-level evidence has been scored. Moreover, both representation methods can be used when either the HTML posts or the summary information from the XML feeds is indexed.

Hence, in this paper, we investigate the blog distillation task. An effective blog search engine will provide relevant key blogs with central interest in the topic areas. We investigate how to provide an effective blog search engine from two angles. Firstly, is it sufficient for a blog search engine to only index the summary information from the XML feed for each blog, or should each permalink post be downloaded and indexed? Secondly, for both adopted indexing strategies, we explore how blogs should be interpreted for effective ranking. In particular, we investigate two interpretations: considering the blog as a whole entity; or considering it as an aggregate of its posts. Finally, we investigate how techniques such as clustering, cohesiveness and date-related evidence can be used to identify the central interest topic area of each blog with the aim to enhance retrieval effectiveness. Our experiments are carried out using the blog distillation task test collection created at the TREC 2007 Blog track [19].

The expert search task, in which experts in an enterprise organisation are ranked with respect to their predicted expertise about a query. In a similar manner fashion to ranking blogs, an expert's expertise can be viewed as the aggregate of their publications. The task has been studied in the TREC Enterprise track.

The contributions of this paper are three-fold: Firstly, we investigate the connections between the fairly well studied expert search task and the new task of blog distillation. Secondly, we explore the retrieval performance impact of indexing only the XML feeds of the blogs, compared to indexing the HTML permalink document of each blog post. Thirdly, we investigate how several intuitions about blog-specific evidence can be modelled and integrated into the proposed retrieval approach, such as the central interests of a blogger, whether an interest is recurring, and whether the blogger has a coherent blogging language.

The remainder of this paper is structured as follows. In Section 2 we introduce the blog distillation task of TREC 2007. Section 3 discusses how aggregates can be ranked in response to a query. We detail the experimental setup we apply in this work in Section 4, and provide experimental results in Section 5. In Section 6, we propose and evaluate a ranking extension that takes into account the number of posts in the blog. Section 7 investigates how bloggers with a central or recurring interest in the topic area can be identified. In Section 8, we apply other techniques to improve the retrieval effectiveness of our system. We provide concluding remarks in Section 9.

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0"
 xmlns:content="http://purl.org/rss/1.0/modules/content/"
 xmlns:dc="http://purl.org/dc/elements/1.1/">
<channel>
 <title>lixo.org</title>
 <link>http://www.lixo.org</link>
 <description>letting the problem solve itself</description>
 <pubDate>Tue, 22 Nov 2005 22:40:36 +0000</pubDate>
 <item>
  <title>London Everything Meetup</title>
  <link>http://www.lixo.org/archives/2005/11/22/london-meetup/
  </link>
  <pubDate>Tue, 22 Nov 2005 19:45:24 +0000</pubDate>
  <dc:creator>Carlos Villela</dc:creator>
  <description> It looks like we're having a Christmas party
   at the Old Bank of England
...
```

Figure 1: An example RSS feed from a blog in the TREC Blog06 test collection. Structured information is provided about the blog (lixo.org), and one or more posts (the first titled London Everything Meetup).

2. BLOG RETRIEVAL AT TREC

The TREC Blog track was initiated in TREC 2006 with the aims of investigating information access in the blogosphere, and providing test collections for common information seeking tasks in the blogosphere setting [19, 22]. Since then, both context and concept queries have been investigated within the TREC setting. The blog distillation task, which investigates concept queries, first ran in TREC 2007.

As mentioned in Section 1, a popular feature of blogs is that with each blog is associated an XML feed, which is updated each time a new post is made to the blog. Many online and offline tools exists for users to read the postings of all the blogs they subscribe to in one interface (known generally as RSS readers). The XML feeds are also used by blog search engines, to enable them to obtain a list of all the new posts to a blog, and hence significantly reduce both their bandwidth usage for crawling and computing resources for indexing.

Two common formats for XML feeds exist: Really Simple Syndication (RSS), and Atom Syndication Format (commonly known as Atom). Figure 1 gives an example of an RSS XML feed for a blog. Within each item of the feed, there is a link to the HTML post *permalink* document, as well as the title and a description of the content of the post (we denote the title and description information the *XML content* of each post). The HTML permalink document contains the full post and any reader comments. However, while the description in the RSS feed can contain the entire text of the blog posting, many feeds only provide a few paragraphs - enough to whet the appetite of a user reading the blog via their RSS reader, who can then follow the link to the permalink to read the full post. There can be various reasons for this succinctness, such as the blogger wants to drive users to his blog so he/she can gain revenue from context advertising. Alternatively, if the full content is given, spammers may automatically republish the blog on another site, in order to gain advertising revenue [12].

For the purposes of the Blog track, TREC created a new Web test collection called Blog06, based on a repeating crawl

Quantity	Value
Number of Unique Blogs	100,649
RSS	62%
Atom	38%
First Feed Crawl	06/12/2005
Last Feed Crawl	21/02/2006
Number of Feeds Fetches	753,681
Number of Permalinks	3,215,171
Feeds (Uncompressed)	38.6GB
Permalinks (Uncompressed)	88.8GB

Table 1: Salient statistics of the Blog06 collection, including both the XML feeds and HTML permalink posts components.

of a set of blogs [15]. In particular, the collection was created by monitoring the RSS or Atom XML feeds of over 100,000 blogs for 11 weeks, and after a two week delay, downloading the blog posts (known as permalinks). The purpose of the two week delay was to allow any comments on the blog post to be collected. Table 1 details the salient statistics of the TREC Blog06 test collection. Both XML feeds and HTML permalinks were provided in the Blog06 test collection, to allow Blog track participants to experiment with both sources of evidence.

The TREC 2007 blog distillation task was created along similar lines to other existing TREC tasks [19]. A test collection was created that mimics, within a repeatable experimental setting, the blog distillation task, where users are looking for new blogs of interest to them, to add them to their RSS readers. Systems were asked to identify key blogs, which exhibit a principle recurring interest in the query. In particular, queries (known as topics) were contributed by the TREC participants. All participating systems then gave their rankings of blogs for each query, which were then pooled for the relevance assessing phase. Participating groups were responsible for the relevance assessing of the pooled blogs for the topics they proposed. When assessing the relevance of a blog, the assessors were asked to read as many or as few posts of the blog as they wish, before making an informed choice of the relevance of the blog as a whole, i.e. *whether the blog is principally devoted to the topic and would be recommended to subscribe to as an interesting feed about the topic area* [19].

For a blog search system, the repeated crawling of blogs is made easier by the provision of XML feeds, which list the URLs of new posts and a summary of their content. An obvious question that arises is whether retrieval using only the XML feed is effective enough for an accurate search system, or whether each HTML post (permalinks) needs to also be downloaded to ensure good retrieval performance at cost of additional crawler bandwidth and indexing time. In the TREC paradigm, this corresponds to developing a system that indexes the feeds component of the Blog06 collection, or the permalinks component, respectively.

Consider that each blog is represented in the Information Retrieval (IR) search system as a large *virtual document* containing all XML content for each blog post seen thus far by the system. An easy way to then rank blogs in response to a query would be simply to rank these virtual documents directly. Alternatively, if the blogs are indexed using their composing posts, then we have to find a way to compute a score for the blog based on a scoring of its constituent posts. Inspired by an expert search approach, which adapts data fusion techniques to rank candidate experts as the aggregate of their expertise-representing documents [16] (e.g. their publications), in this work we investigate the connection between expert search and blog distillation. In the next two sections, we describe both indexing and ranking strategies.

3. RANKING AGGREGATES

The aim of a blog search engine is to identify blogs which have a recurring interest in the query topic area. Our intuitions for the blog distillation task are as follows: A blogger with an interest in a topic will blog regularly about the topic, and these blog posts will be retrieved in response to a query topic. Each time a blog post is retrieved for a query topic, then it can be seen as an indication (a vote) for that blog to have an interest in the topic area and thus more likely that the blog is relevant to the query. This task is then very similar to the expert search task, in that both tasks aggregate the documents that are ranked in response to a query. In particular, a candidate's expertise can be interpreted as the aggregate of their (e.g.) publications, and likewise a blog's interest can be interpreted from the aggregate of all its constituent posts.

In this work, we use the adaptable Voting Model for Expert Search [16]. In this model, candidate experts are ranked by examining the ranking of documents with respect to the query. If an expert has many associated documents highly ranked in the ranking of documents, then these are seen as votes that they be ranked higher than another expert with less or lower ranked documents. Indeed, Macdonald & Ounis proposed 12 voting techniques in [16], based on adaptations of common data fusion techniques, such as CombSUM, CombMNZ or BordaFuse. Many of the voting techniques were shown to perform well on the TREC Enterprise track expert search task when applied using several statistically different document weighting models [18].

Following the good performance of the voting techniques proposed by Macdonald & Ounis, we use four representative techniques in this work as they apply various sources of evidence from the underlying ranking of blog posts.

In the simplest technique, called Votes, blogs are ranked by the number of their posts ranked in response to a query. In particular, the retrieval score for a blog B with respect to a query Q, denoted $score(B, Q)$ is:

$$score_{Votes}(B, Q) = \|R(Q) \cap posts(B)\| \quad (1)$$

where $R(Q)$ is the underlying ranking of blog posts, and $posts(B)$ is the set of posts belonging to blog B. Note, that in contrast with the expert search task where a document can be associated to more than one candidate (e.g. a publication with multiple authors), in the blog setting, each post is associated to exactly one blog.

Next, the CombMAX voting technique scores a blog B by the retrieval score of its most highly ranked post:

$$score_{CombMAX}(B, Q) = max_{\substack{p \in R(Q) \\ \cap\ posts(B)}} (score(p, Q)) \quad (2)$$

where $score(p, Q)$ is the retrieval score of blog post p as computed by a standard document weighting function. The set $R(Q) \cap posts(B)$ is the set of retrieved posts belonging to blog B.

Next, the expCombSUM technique ranks each blog by the sum of the relevance scores of all the retrieved posts of the blog, and strengthens the highly scored posts by applying the exponential ($exp()$) function (strong votes evidence):

$$score_{expCombSUM}(B,Q) = \sum_{\substack{p \in R(Q) \\ \cap\, posts(B)}} exp(score(p,Q)) \quad (3)$$

Lastly, the expCombMNZ technique is similar to expCombSUM, except that the count of the number of retrieved posts is also taken into account (number of votes evidence):

$$score_{expCombMNZ}(B,Q) = \|R(Q) \cap posts(B)\| \cdot \sum_{p \in R(Q) \cap posts(B)} exp(score(p,Q)) \quad (4)$$

where $\|R(Q) \cap posts(B)\|$ is the number of posts of blog B that are retrieved in the ranking $R(Q)$.

Note that this aggregate retrieval strategy based on the ranking of blog posts can be applied to both retrieval using the only XML content for each post, or using the HTML permalink documents.

For a baseline ranking strategy, a simple and intuitive way of ranking blogs is the virtual document approach, whereby each blog is represented by a large virtual document containing all term occurrences from all of its constituent posts (either permalink content or XML content) concatenated together. These virtual documents can then be directly ranked in response to a query. This approach was first proposed for the expert search task by Craswell et al. [5].

Other work of note is the formal language models of Balog et al. [3]. In particular, their Model 1 is a formalisation of the Craswell virtual document approach, while in their Model 2, the probability of a candidate is calculated using the sum of the probability of each document with respect to the query, multiplied by the degree of association between the document and candidate. Model 2 has similarities to the CombSUM and expCombSUM voting techniques of Macdonald & Ounis [16], in that all techniques are based on the sum of some measure of how much the document is about the query. Moreover, Balog et al. found that Model 2 is usually superior to the Model 1 (virtual document) approach on the expert search task.

4. EXPERIMENTAL SETUP

As discussed above, we have two forms of alternative content that can be indexed for each post (the XML content, and the HTML permalinks). Moreover, the two alternative ranking strategies - voting techniques and virtual documents - require different index formats. Hence we index the Blog06 collection in four ways:

1. Using a virtual document for all the HTML permalink posts associated to each blog.
2. Using a virtual document for all the XML content associated to each blog.
3. Using the HTML permalink document for each blog post, as a separate index entity.
4. Using the XML content for each blog post as a separate index entity.

For approaches 3 and 4, we use the voting techniques introduced in Section 3 to convert the ranking of blog posts into a ranking of blogs, while for approaches 1 and 2, blogs are scored and ranked directly. Note that when indexing XML feeds, the XML content for a blog post is indexed only once - i.e. on the first occurrence of that post in the feed, and not in subsequent fetches of the feed when the same post was still visible.

	Indexed	
Ranking Strategy	XML content	HTML permalinks
Virtual Documents	#Docs: 100,649 #Tokens: 213,093,984	#Docs: 100,649 #Tokens: 2,841,396,389
Voting Techniques	#Docs: 3,215,171 #Tokens: 213,093,984	#Docs: 3,215,171 #Tokens: 2,841,396,389

Table 2: Statistics for the four created indices. #Docs is the number of documents in the index, #Tokens is the number of tokens in the index.

In all cases, we index using Terrier [21], removing standard stopwords and applying Porter's English stemmer. In particular, there are 2,841,396,389 tokens of text found by indexing all the HTML blog post documents, while only 213,093,984 tokens are found when indexing the XML feeds. There is an order of magnitude difference in the amount of textual content obtained from either source, demonstrating how many bloggers are choosing not to provide full content in their XML feeds, for the reasons described in Section 2. Table 2 gives an overview of the statistics of the four indices.

We rank index entities (whether virtual documents or posts) using the new DFRee Divergence from Randomness [1] (DFR) weighting model, provided in the open source version of Terrier 2.1[1]. This new weighting model is parameter free, and performs effectively on various test collections without the need for any parameter tuning [28]. In particular, we score an entity e (i.e. a blog or a blog post) with respect to query Q as:

$$score(e,Q) = \sum_{t \in Q} qtw \cdot tf \cdot \log_2 \frac{post}{prior} \cdot ((tf+1) \cdot log_2(post \cdot \frac{TFC}{TF} - tf \cdot \log_2 prior \cdot \frac{TFC}{TF}) + 0.5 \cdot \log_2 \frac{post}{prior}) \quad (5)$$

where $prior = \frac{tf}{length}$, $post = \frac{tf+1}{length+1}$, $length$ is the length in tokens of entity e, tf if the number of occurrences of term t in e, TF is the number of occurrences of term t in the collection, and TFC is the number of tokens in the entire collection. Indeed, DFRee has no term frequency normalisation parameter that requires tuning, as this is assumed to be inherent to the model. Hence, by applying DFRee, we remove the presence of any term frequency normalisation parameter in our experiments, while still having a strong baseline. Indeed, applying other DFR models, such as PL2 [1], which contain term frequency normalisation parameters may lead to better retrieval effectiveness, but these would required appropriate training of their parameters.

All our experiments are conducted using the TREC 2007 Blog track, blog distillation task. In particular, this task has 45 topics with blog relevance assessments [19]. While the topic provides the traditional TREC title, description and narrative fields, for our experiments we use the most realistic title-only setting. Moreover, the official ranking of systems in TREC 2007 was done by title-only systems. An example topic is shown in Figure 2. Retrieval performance is reported in terms of Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Precision @ rank 10 (P@10).

5. EXPERIMENTAL RESULTS

In our experiments, we aim to draw conclusions on several points: Firstly, can indexing using only the textual con-

[1] `http://ir.dcs.gla.ac.uk/terrier`

```
<top>
<num>Number: 985</num>
<title>solaris</title>
<desc> Description:
 Blogs describing experiences administrating the Solaris operating
 system, or its new features or developments.
</desc>
<narr> Narrative:
 Relevant blogs will post regularly about administrating or using
 the Solaris operating system from Sun, it's latest features or
 developments. Blogs with posts about Solaris the movie are not
 relevant, not are blogs which only have a few posts Solaris.
</narr>
</top>
```

Figure 2: Blog track 2007, blog distillation task, topic 985.

	MAP	MRR	P@10
From XML feed			
Virtual Documents	**0.2163**	0.5404	**0.4022**
Votes	0.1720<	0.5589	0.3556
expCombMNZ	0.1710≪	**0.6006**	0.3667
expCombSUM	0.1397≪	0.5201<	0.2844≪
CombMAX	0.1011≪	0.4083≪	0.1933≪
Entire Posts			
Virtual Documents	0.1436≪	0.4598≪	0.2778≪
Votes	0.2348	0.5778≪	0.4489
expCombMNZ	**0.2584**	0.7747	**0.4667**
expCombSUM	0.2312≪	**0.7989**	0.4356<
CombMAX	0.1750≪	0.6006≪	0.3356≪

Table 3: Experimental results comparing the virtual document and voting technique approaches, combined with indexing feed or permalink posts. The best result for each index form is emphasised, and statistically significant degradations (calculated using the Wilcoxon matched-pairs signed rank test) from the best are denoted $<$ and $\ll$ for ($p <= 0.01$) and ($p <= 0.05$), respectively.

tent from the XML feeds be as effective as using the full content from the HTML permalinks blog posts; Secondly, which ranking strategy is most effective for ranking blogs - virtual documents versus voting techniques; and lastly, given that we experiment with various possible voting techniques, whether there is any variance between the techniques.

The observed results are provided in Table 3. The table details the approaches of both the virtual document and voting techniques forms of ranking, applied when using either the XML content or the full HTML permalinks for the textual content of the blog posts.

On analysing Table 3, we can draw several conclusions. Firstly, there is a marked overall difference in retrieval performance for indexing feeds versus blog posts. Indeed, the highest performance achievable using the XML content is 0.2163 MAP, while 0.2584 is achievable when the entire post has been indexed. Given the difference in number of indexed tokens between the two sources, we suggest that it is surprising that this difference is not greater. However, on applying the Wilcoxon matched-pairs signed rank significance test (not shown in the table), we note that the MAP differences between each voting technique setting on XML content, and the equivalent setting on permalinks is statistically significant (one exception is the Votes technique) in favour of those applied using the permalink content, establishing that the best retrieval effectiveness can be found using voting techniques on the entire posts.

Comparing the voting techniques with the virtual document approach, it is apparent that the virtual document approach performs better than the voting techniques for the reduced content from the XML feeds - indeed, from Table 3, we note that this is significant for all MAP cases, and for some MRR and P@10 cases. A notable exception is the expCombMNZ technique which is best for MRR, but not significantly better than the virtual document approach.

However, when the full blog posts are indexed, the virtual document approach significantly under-performs, and is unable to achieve the retrieval performance of some approaches on even the XML feed content. In contrast, the opposite is observed for the voting techniques: while these do not perform well on the XML feed content, they provide excellent performance on the full permalink content. This is similar to the observations of Balog et al. in their comparison of Model 1 and Model 2 expert search models [3].

We suspect these differences of performance can be explained by the fact that the virtual document approach does not weight individually the contribution of each blog posts weight to the blog's likely relevance, and hence will struggle to identify informative content in the large virtual documents from the full blog posts. For the XML content, the average size of the virtual document is much smaller, with often only a few sentences contributed from each blog post. In this scenario, the mean virtual document length is actually similar to the mean HTML blog post length in the collection, meaning that the weighting model is able to differentiate easier between relevant and irrelevant blogs. It is of note that without an explicit document length normalisation component, it is impossible to tune the DFRee model to any one setting.

Comparing the voting techniques, the expCombMNZ techniques seems to perform best overall, mirroring previous studies by Macdonald & Ounis in the expert search setting [16]. The surprising performance of the simple Votes technique suggests that simply counting the number of on-topic blog posts is a good indicator of the likely relevance of the blog. This is intuitive, as the more a blogger blogs about the topic area, the more likely that they have a recurring interest in the topic area, and that a user would find the blog interesting to subscribe to.

The CombMAX technique, which only considers the top-ranked post for each blog, is less suitable in this task, as a blog which only contains one on-topic post will be highly ranked, when they do not necessarily exhibit the recurring interest in the topic area. The response of the blogosphere to the London terrorist bombings was examined in [29], and this provides examples of why CombMAX is not effective: many bloggers made posts about the London bombings, but these blogs would not be relevant to a query about 'terrorism and security' as a blog with a really recurring and central interest in the topic would be. This contrasts with the usage of CombMAX in the expert search task, where a (e.g.) publication which is very much about the query topic is likely to be an excellent indicator of expertise of a candidate.

Overall, we conclude that it appears that the full HTML content of each blog post should be downloaded and indexed for a blog search engine to achieve the highest retrieval performances. Using the voting techniques for ranking documents provides the best performance, and hence we will use only this approach for the remainder of this paper.

The best result achieved (MAP 0.2584) would have been ranked as third group in the TREC 2007 Blog distillation task [19], and this was achieved without the training of any parameters - indeed all models used thus far have been completely parameter-free. The results here would likely be improved by using additional features, such as other weighting models (including field-based weighting models and other weighting models with tunable parameters, e.g [7]), proximity of query terms, query expansion/collection enrichment and the like, as used by many of the submitted TREC runs [19]. In Section 8, we compare further to other TREC 2007 submitted runs to demonstrate the achievable retrieval performance under similar settings.

Moreover, given the experimental results found above, particularly the importance of the number of on-topic posts, we wonder if it would be possible to improve the voting techniques by introducing a parameter that controls the influence of the blog size on the chance of the blog receiving a vote. In doing so, we would remove any unfair bias toward blogs with many posts, which are likely to be retrieved because a blog post contained a query term randomly. Section 6 investigates the application of normalisation to improve the retrieval performance of the voting model. In contrast, in Section 7, we investigate further techniques to determine if the query topic is one of the main interests of the blogger.

6. BLOG SIZE NORMALISATION

Similar to the previous work in expert search, the best performing voting technique appears to be the expCombMNZ technique. However, an issue using such a technique is that prolific bloggers may gain an unfair advantage in the ranking. This is because the more a blogger writes, the more likely a query term will appear at random in a blog post (for example, many blog posts contain links to other recent posts, with the title of each post identical to the link anchor), and hence the blog will receive extra erroneous votes. In contrast, the relevance of a blog is more likely to be related to the quality of the on-topic posts, and not whether the blogger is prolific or not.

To this end, inspired by the work done previously in the IR field with regard to document length normalisation, we investigate how the importance of a vote can be normalised and controlled by the number of posts in the blog. Similarly, document length normalisation has been investigated in various literature [8, 24, 26] with a view to controlling the chance that a document is retrieved based on its size. In this work, we adapt a classical document length normalisation technique, *Normalisation 2*, from the Divergence from Randomness framework, and integrate it into the voting model, as Normalisation 2 was shown to be an effective approach for document length normalisation [1]. The score of a blog is adapted as follows:

$$score_{Norm}(B,Q) = score(B,Q) \cdot log(1 + c \cdot \frac{avg\_l}{l}) \qquad (6)$$

where c is a free parameter ($c > 0$), l is the number of posts in blog B, and avg_l is the average number of posts for all blogs. Note that we can also measure the size of a blog, l, in terms of the number of tokens in its constituent posts in B, and avg_l as the average number of tokens associated to each blog. In particular, we denote Norm2p as the normalisation applied using the number of posts in the blog, while Norm2t denotes when the normalisation is applied using the number of tokens in the blog.

Training		c	MAP	MRR	P@10
From XML feed					
	expCombMNZ	-	0.1710	0.6006	0.3667
(Default)	+ Norm2t	1	0.1913≫	0.6173	0.3933
(Train)	+ Norm2t	0.04	0.1926≫	0.6396	0.4044≫
(Test)	+ Norm2t	0.12	0.1934≫	**0.6402**	0.4067≫
(Default)	+ Norm2p	1	0.1939≫	0.6135	0.4156≫
(Train)	+ Norm2p	1.36	0.1932≫	0.6109	0.4089≫
(Test)	+ Norm2p	0.04	**0.1970**≫	0.6176	**0.4533**≫
	expCombSUM	-	0.1397	0.5204	0.2844
(Default)	+ Norm2t	1	0.1489	0.5145	0.3133
(Train)	+ Norm2t	8.88	0.1524>	0.5138	**0.3244**≫
(Test)	+ Norm2t	10.91	**0.1528**≫	0.5254	**0.3244**≫
(Default)	+ Norm2p	1	0.1497>	0.5565	0.3222≫
(Train)	+ Norm2p	12.06	0.1513≫	0.5578	0.3178≫
(Test)	+ Norm2p	6.60	0.1520≫	**0.5615**	0.3222≫
Entire Permalink Posts					
	expCombMNZ	-	0.2584	0.7747	0.4667
(Default)	+ Norm2t	1	0.2744≫	**0.8244**	0.5089≫
(Train)	+ Norm2t	8.18	0.2703≫	0.7964	0.5000≫
(Test)	+ Norm2t	0.90	0.2746≫	0.8235>	0.5111≫
(Default)	+ Norm2p	1	0.2852≫	0.8226>	0.5200≫
(Train)	+ Norm2p	0.29	0.2877≫	0.8226>	**0.5267**≫
(Test)	+ Norm2p	1.52e-4	**0.2902**≫	0.8226>	0.5244≫
	expCombSUM	-	0.2312	0.7989	0.4356
(Default)	+ Norm2t	1	0.2410	0.8756	0.4422
(Train)	+ Norm2t	4.20	0.2422	0.8542>	0.4511
(Test)	+ Norm2t	2.84	0.2425≫	0.8559	0.4489≫
(Default)	+ Norm2p	1	0.2588≫	**0.8772**>	0.4822≫
(Train)	+ Norm2p	1.57	0.2571≫	0.8643	0.4800≫
(Test)	+ Norm2p	0.28	**0.2603**≫	0.8754>	**0.4844**≫

Table 4: Experiments using Blog Size Normalisation. Best settings for each measure, voting technique and index form are emphasised; statistically significant increases from the voting technique without normalisation applied are denoted > and ≫ for ($p <= 0.05$) and ($p <= 0.01$), respectively. Note that the baseline applications of expCombSUM and expCombMNZ do not have a c parameter.

We test the proposed normalisation technique using both indices in combination with the expCombMNZ and expCombSUM baseline techniques. Moreover, because this is a new task in TREC, there is not much training data on which to find a good setting for the normalisation parameter c. Therefore, we experiment with a default setting of $c = 1$ as suggested by Amati [1]. In addition, we create seven training queries with shallow relevance assessments. However, as these are not necessarily representative of the test set, we also provide the ideal setting where we assume that optimal training was available. This means that in this last setting, we have directly optimised the parameter using the test set for training. Training is performed using simulated annealing processes to maximise MAP [11].

Table 4 presents the results of our normalisation experiments. Analysing the table, we can draw several conclusions. Firstly, that applying normalisation can improve the retrieval performance of the voting techniques on both the XML and the HTML permalink content. For the permalink content, marked increases are apparent, which are often statistically significant. Similarly, for the XML content, there are often significant increases for MAP and P@10, for both voting techniques. On comparing expCombMNZ with expCombSUM, it is apparent that expCombMNZ performs better, regardless of the normalisation applied, on most measures (one exception is MRR for permalinks content).

Of the three settings for the c parameter (default, trained on training queries, optimal training), we note only small differences in retrieval effectiveness between each of the three settings, and conclude that the normalisation is not overly

sensitive to the c parameter setting. However, as the parameter settings were trained to maximise MAP, in some cases other measures are impaired compared to the default parameter setting. Finally, comparing the Norm2p and Norm2t methods of normalisation, we note that overall Norm2p performs slightly better, inferring that counting the size of a blog using its number of posts is best. This is probably explained in that the number of tokens in each post is already taken into account by DFRee when ranking posts. Overall, we conclude that the introduction of normalisation to voting techniques allows them to be adapted to take a more refined view of the number of votes for each blog, by ensuring that blogs with many posts do not gain an unfair bias in the final ranking - users do not necessarily prefer prolific bloggers that blog about many topics including their topic of interest over bloggers that blog more continuously on the topic of interest.

7. CENTRAL & RECURRING INTERESTS

So far, we have been experimenting within the framework of the voting model applied to rank blogs. Now, we wish to investigate blog-specific features that allow us to separate the key relevant blogs from the rest. In particular, we test several retrieval enhancing techniques that aim to boost blogs for which the blogger has shown a central or recurring interest in the topic area. In doing so, we aim to model more fully the definition of a relevant blog given to the assessors (as described in Section 2). In this respect, we form three hypotheses:

- **Central Interest**: If the posts of each blog are clustered, then relevant blogs will have blog posts about the topic in one of the larger clusters.
- **Recurring Interest**: Relevant blogs will cover the topic many times across the timespan of the collection.
- **Focused Interest**: Relevant blogs will mainly blog around a central topic area - i.e. they will have a *coherent* language model with which they blog.

In the remainder of this section, we detail each hypothesis in turn and then provide the results and analysis.

7.1 Central Interests

Some bloggers may have a wandering attention span, blogging about many topics. For instance, a primarily technical blog may occasionally post in response to a real-world event, or comment on a personal or off-topic aspect. For example, in Thelwall's work about the London bombings [29], it was noted that the bombings had a noticeable impact on the blogosphere in July 2005. However, it is of course obvious that not all of these blogs were interested in terrorist and security issues before this day, and consequently their interest in the London events would fade with time. In this work, we desire to identify blogs which not only contain mainly relevant posts to the topic area, but where the blogger primarily blogs in the topic area of the query.

To achieve this, clustering seems to be a good option. We cluster the set of posts associated to each blog, hoping that clusters will form, which represent the main topic areas of each blog. In particular, in this paper we apply a single-pass clustering algorithm [23] to cluster all the posts of the blogs with more than θ posts. This process is done offline at indexing time. In the clustering, the distance function is defined as the Cosine between the average of each cluster. The clusters obtained are then ranked by the number of documents they contain - the largest clusters are representatives of the central interests of the blog. In particular, we form a *quality score*, which measures the extent to which a blog post is central to a blogger's interests, by determining which cluster the post occurs in. This is calculated as follows:

$$Qscore_{Cluster}(p, B) = \frac{1}{cluster(p, B)} \quad (7)$$

where $cluster(p, B)$ is the rank of the cluster in which post p occurred for blog B (largest cluster has rank 1). The above integration of central interest evidence into the voting technique strengthens votes from documents which are found in larger clusters of posts, because the largest clusters are assumed to represent the blogger's main interest areas. Moreover, if no clustering has been applied for the blog (i.e. the blog has less than θ posts), then $Qscore_{Cluster}(p, B) = 0$. We integrate the clusters quality score with the expCombMNZ voting techniques for scoring a blog to a query in Equation (8) below. Note that while it may be possible to adapt other voting techniques to integrate the clusters quality score evidence, for our experiments we focus solely on expCombMNZ.

$$score_{expCombMNZ.Cluster}(B, Q) = \|R(Q) \cap posts(B)\| \cdot \sum_{\substack{p \in R(Q) \\ \cap posts(B)}} exp(score(p, Q) + \omega \cdot Qscore_{Cluster}(p, B)) \quad (8)$$

In this work, we use the default setting of $\theta = 1$ - i.e. we only skip blogs which have one or zero posts. In these cases, blogs with only a single post cannot be checked to have a central interest, as only at most one post represents their interest to the system.

7.2 Recurring Interests

If a blogger has an interest in a topic area, it is likely that they will continue to blog about the topic area repeatedly and frequently. Indeed, the definition for a relevant blog in the blog distillation task gives a clue that the timing of on-topic posts by a blog may have an impact on the overall relevance of the blog. In particular, we believe that a relevant blog will continue to post relevant posts throughout the timescale of the collection.

With this in mind, we break the 11 week period of the Blog06 collection into a series of DI equal intervals (where DI is a parameter). Then for each blog, we measure the proportion of its posts from each time interval that were retrieved in response to a query. We define a $Qscore_{Dates}(B, Q)$ for each blog B as follow:

$$Qscore_{Dates}(B, Q) = \sum_{i=1}^{DI} \frac{1 + \|R(Q) \cap dateInterval_i(posts(B))\|}{1 + \|dateInterval_i(posts(B))\|} \quad (9)$$

where $dateInterval_i(posts(B))$ is the number of posts of blog B in the ith date interval. Note that we smooth this probability distribution using Laplace smoothing to combat sparsity problems (e.g. when a blog had no posts in a date interval). We integrate the $Qscore_{Dates}(B, Q)$ evidence as:

$$score(B, Q) = score(B, Q) \times Qscore_{Dates}(B, Q)^{\omega} \quad (10)$$

where $\omega > 0$ is a free parameter. We use $DI = 3$, which approximates the month where the post was made (the corpus

timespan is 11 weeks). Initial experiments found that using higher values for DI does not change the results, due to the timespan of the corpus. Finally, note that as this evidence requires knowledge of the ranking of posts for a query, it has to be calculated during the retrieval phase, but without adding high overheads.

7.3 Focused Interests

We believe that relevant blogs will likely be blogs for which the topic area is a main interest of the blog, and the blog will not digress onto other topics excessively. Statistically, this can be measured by examining the *cohesiveness* of the language model of the set of blog posts. Indeed, cohesion has been investigated in cluster analysis, with the view to ensuring that a cluster models a coherent set of documents [27].

In [17], Macdonald & Ounis examined three measures of cohesiveness, within the context of query expansion for expert search. A measure of cohesiveness examines all the documents associated with an aggregate, and measures on average, how different each document is from all the documents associated to the aggregate. In this work, the cohesiveness of a blog feed B can be measured using the Cosine measure from the vector-space framework as follows:

$$Cohesiveness_{Cos}(B) = \frac{1}{\|posts(B)\|} \cdot \sum_{p \in posts(B)} \frac{\sum_{t \in posts(B)} tf_p \cdot tf_B}{\sqrt{\sum_{t \in p} (tf_p)^2} \sqrt{\sum_{t \in posts(B)} (tf_B)^2}} \quad (11)$$

where $posts(B)$ denotes the set of blog posts associated with blog feed B. Moreover, tf_p is the term frequency of term t in post p, and tf_B is the total term frequency of term t in all posts associated with blog B (denoted $t \in posts(B)$) - i.e. this is similar to the centroid of a cluster. $Cohesiveness_{Cos}$ measures the mean divergence between every document in the blog and the blog itself. Note that $Cohesiveness_{Cos}$ is bounded between 0 and 1, where 1 means that the posts have a completely cohesive language model. We integrate the cohesiveness score with the $score(B, Q)$ for a blog to a query as follows:

$$score(B,Q) = score(B,Q) + log(1 + \omega \cdot Cohesiveness_{Cos}(B)) \quad (12)$$

where $\omega > 0$ is a free parameter. Similar to the clustering approach proposed in Section 7.1, cohesiveness can be calculated offline for each blog at indexing time.

7.4 Results & Analysis

Here, we test the proposed central interest features described above. We test only using the expCombMNZ voting technique using permalink content, as this is the best setting achieved thus far. The results of our experiments are detailed in Table 5. In particular, we report two settings for Cohesiveness, namely when the blog cohesiveness is calculated on the HTML permalink content, and when the blog cohesiveness is calculated on the XML content. We believe that using the XML content will reduce the amount of noise introduces by the boilerplate HTML in each permalink blog post. As in Section 6, we report the evaluation measures when the settings are obtained using the sparse training data (with only seven queries), and when the settings are trained using the test data.

On analysing the results in Table 5, we make several observations: Firstly, the Dates feature is the most promising, resulting in statistically significant improvements in both MAP and P@10, even when using the sparse training data. Using optimal training, even results in a further increase - as high as 0.2980 MAP. In essence, the proposed Dates feature successfully modelled a notion of recurrence required by the blog distillation task.

Next, the Clusters approach also results in statistically significant improvements in MAP, reaching a high of 0.2654 MAP, suggesting that this technique has potential for identifying the central interests of each blogger. Further investigation of this technique may focus on using different clustering techniques or similarity functions.

Unexpectedly, the cohesiveness measures do not result in increased retrieval performance. In general, the trained parameter value for ω is typically very small, indicating that the optimisation process is recommending that the feature should not be applied. As discussed above, we calculated the cohesiveness measure on two indices, from the XML content and the permalink content, to assess whether the noise introduced by the HTML boilerplate is behind the disappointing performance of the cohesiveness measure. While the cohesiveness measures calculated on the XML content performs better when trained on the training queries, for the optimal setting there is little difference between the measures calculated on the different indices. Perhaps other methods of combining this and the other central and recurring interest sources of evidence with the voting technique retrieval score should be investigated to assess the full usefulness of these sources of evidence.

Overall, we conclude that the Dates and Clusters features are good evidence, which seem to have encompassed some aspects of the task, namely the centrality of the query topic to the blog, and the recurrence aspect. One would expect that combining these two features is a promising prospect, however such a combination is not straightforward. In the future, we would like to investigate fully a uniform approach of integrating such evidences, when more training data becomes available.

8. ENHANCING RETRIEVAL PERFORMANCE

In comparison with the participating systems in the TREC 2007 task, the best results reported so far would have ranked between first and second groups for automatic title-only runs [19]. In this section, we apply techniques to increase the retrieval performance of our system.

Firstly, we apply a field-based weighting model. Various Web IR studies have shown that taking into account the different frequencies of query terms in the title, body of each blog post, and in the anchor text of incoming hyperlinks to the post, can have an impact on the retrieval performance of a system. We adapt DFRee to be a field-based weighting model in a similar manner to Robertson et al. [25], and denote the new weighting model as DFReeF. In DFReeF, the term frequency tf is computed as $tf = \sum_f w_f \cdot tf_f$. Hence, tf is the weighted sum of the term frequencies of term t in each field f. $w_f > 0$ are weights that control the influence of each field in the ranking. We train w_f using the training dataset with seven queries described earlier.

As discussed earlier, other weighting models that contain length normalisation components that can be tuned may be better suited for effective retrieval performance. For this reason, we also show results using PL2 [1] and its field-based derivative PL2F [14]. In PL2F, field term frequencies are

	Train/Test				Test/Test			
Approach		MAP	MRR	P@10		MAP	MRR	P@10
expCombMNZ	-	0.2584	0.7747	0.4667	-	0.2584	**0.7747**	0.4667
+ Clusters	$\omega = 8.9$	0.2628>	0.7624	0.4844	$\omega = 4.02$	0.2654≫	0.7665	0.4822
+ Dates	$\omega = 0.48$	**0.2788**≫	**0.7893**	**0.5022**≫	$\omega = 3.49$	**0.2980**≫	0.7707	**0.5289**≫
+ Cohesiveness (HTML)	$\omega = 1.4$	0.1847≪	0.7719	0.3556≪	$\omega = 0.003$	0.2577	**0.7747**	0.4733
+ Cohesiveness (XML)	$\omega = 0.0035$	0.2280≪	0.7746	0.4556	$\omega = 7.34e-5$	0.2532	**0.7747**	0.4733

Table 5: Results for Section 7. Train/Test denotes when the parameter setting is trained on a training set, while Test/Test denotes when the parameter is trained using the test set of topics. Significant increases over expCombMNZ are denoted $>$ ($p \leq 0.05$) and $\geq$ ($p \leq 0.01$), while significant decreases are denoted $<$ and $\ll$.

combined after each frequency has been normalised with respect to the average length of that field in all documents of the collection. This allows the contribution of the term occurrence in a field to be properly weighted. In this work, we use the parameter setting suggested in [7] for opinion finding on the Blog06 collection.

Secondly, we take into account dependence and proximity of query terms in blog posts to increase the retrieval effectiveness of the blog distillation search system. We use the DFR pBiL2 model to weight the occurrences of pairs of query terms that appear within a given number of terms of each other in the document [14]. pBiL2 is a useful model for term dependence, as it does not consider the frequency of the query term pair in the collection (which can be expensive to compute), instead calculating the informativeness of the query term pair occurring in each document based on the document's length.

Lastly, inspired by [6], we expand the original query by using another collection, known as collection enrichment [13]. This is achieved by performing retrieval on the other collection, expanding and reweighing the query, then using this expanded query to retrieve blog posts. We use the Bo1 term weighting model, which has previously been successfully applied for collection enrichment [9]. In particular, we use a copy of the Wikipedia database from a similar timeframe as Blog06 for collection enrichment. This is a useful source for collection enrichment as it helps to expand the concept queries to contain other related terms using the relevant Wikipedia articles [2, 6]. We use Terrier's default setting of expansion of 10 terms from 3 documents.

In Table 6, we firstly compare the DFRee and PL2 weighting models, together with their field-based equivalents. We can see that while PL2 performs almost identically to DFRee, PL2F markedly outperforms DFReeF. Moreover PL2F statistically outperforms the DFRee.

We now combine the other various features described previously, including Norm2p, Proximity, Dates and collection enrichment, with PL2F (statistical significance with respect to PL2F is shown in Table 6). Where the features contain parameters, we use settings trained on the seven training queries described above. From the results, we note the following: Norm2p continues to show significant improvement when applied to the stronger PL2F ranking of posts; applying Proximity to PL2F + Norm2p does not improve MAP, but does improve MRR. In contrast, applying Dates with Norm2p and Proximity improves MAP and P@10 but not MRR. Collection enrichment shows to be the best performing additional feature, and combination with Dates improves all measures further.

Comparing to the best TREC 2007 submitted runs, we note that our best setting is close to the best submitted automatic title-only runs (MAP 0.3475 vs 0.3695). Moreover, the P@10 and MRR exhibited are markedly higher than any

expCombMNZ	MAP	MRR	P@10
+ DFRee	0.2584<	0.7747	0.4667<
+ PL2 c=2	0.2586	0.7328	0.4667
+ DFReeF	0.2705	0.7764	0.5067
+ PL2F (setting taken from [7])	0.2909	0.7686	0.5222
+ PL2F + Norm2p	0.3174≫	0.7772	0.5733≫
+ PL2F + Norm2p + Proximity	0.3129≫	0.7865	0.5733≫
+ PL2F + Norm2p + Proximity + Dates	0.3187≫	0.7798	0.5800≫
+ PL2F + Norm2p + Enrichment	0.3418	0.8342≫	0.5956≫
+ PL2F + Norm2p + Enrichment + Dates	**0.3481**≫	**0.8405**	**0.6044**≫

Table 6: Applying different document weighting models (PL2 & PL2F), enrichment and proximity features in combination with Blog Size normalisation (Norm2p) and Recurring Interests (Dates). Statistical significance to PL2F is shown.

of the submitted runs to TREC (MRR 0.8405 > 0.8093, P@10 0.6044 > 0.5356) [19]. The high performance of MAP of the best performing group is likely due to the more extensive training they performed. For 8 queries, Elsas et al. manually assessed for relevance the blogs retrieved down to rank 50 by a baseline system [6], and this is used as a training dataset. In contrast, the settings for PL2F applied in this section are those reported in [7] for opinion finding on the same collection. In general, all the proposed features in Sections 6, 7 and 8 would likely be improved given a more suitable and larger training dataset.

Overall, we conclude that the proposed model for key blog distillation can perform effectively, especially for the important high-precision evaluation measures.

9. CONCLUSIONS & FUTURE WORK

In this work, we introduced and motivated the blog distillation task, which recently ran as part of the TREC 2007 Blog track. We investigated the connections between this task and the expert search task, and examined two methods of ranking blogs for a query, namely voting techniques and virtual documents. Moreover, we also explored whether indexing the XML feed of a blog is sufficient for good retrieval performance, or whether the entire HTML permalink should be indexed for each post in a blog. Moreover, we compared and contrasted what usually works on the expert search task with our experimental results on the blog distillation task. In general, we found that the effective models perform well on both tasks.

Our experimental results showed that while indexing only the XML feeds gave a reasonable retrieval performance, this was markedly lower than indexing the full HTML permalink content for each blog post. For a blog search engine, this is an important result, as indexing permalink documents in this setting requires an extra 90GB of content to be downloaded in order to achieve full retrieval effectiveness. For ranking, the voting techniques previously applied in expert search performed well, particularly on the full HTML permalink content.

Next, to remove any bias toward prolific blogs in the search engine ranking, we were inspired by previous work in document length normalisation. We investigated adding a normalisation component to the voting techniques, and found that this could indeed improve the retrieval performance. Finally, we proposed various approaches for identifying the central and recurring interests of a blog with the aim to address the specifics of the blog distillation task. Of the proposed approaches, we can identify the central interests of a blog using clustering, and can identify bloggers with recurring interests in a topic area by the regularity of their relevant posts. Clustering led to a 3% improvement in MAP over the baseline. Recurring interests (Dates) led to a statistically significant improvement of 7% when little training is done, to 15% when a better setting is used.

The best experimental results in this study are extremely competitive and compare well to the current state-of-the-art at TREC, particularly when similar additional features such as collection enrichment and recurring interests (Dates) are applied. Further improvements may be achievable given the availability of appropriate training data - the upcoming TREC 2008 blog distillation task should allow for additional insights to be drawn on the effectiveness of the various blog search techniques. In the future, we would like to broaden our research in this task to cover the analysis of linkage patterns between blogs and how this information can be utilised to enhance the retrieval performance on this task, as well as extracting and utilising tags that bloggers may have added to their posts.

10. REFERENCES

[1] G. Amati. *Probabilistic Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, Univ. of Glasgow, 2003.

[2] J. Arguello, J. Elsas, J. Callan, and J. Carbonell. Document Representation and Query Expansion Models for Blog Recommendation. In *Proceedings of ICWSM 2008*, 2008.

[3] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of SIGIR 2006*, pages 43–50, 2006.

[4] N. Craswell and D. Hawking. Overview of TREC-2004 Web track. In *Proceedings of TREC-2004*, 2004.

[5] N. Craswell, D. Hawking, A.-M. Vercoustre, and P. Wilkins. Panoptic expert: Searching for experts not just for documents. In *AusWeb-2001 Poster Proceedings*, 2001.

[6] J. Elsas, J. Arguello, J. Callan, and J. Carbonell. Retrieval and Feedback Models for Blog Distillation. In *Proceedings of TREC-2007*, 2008.

[7] D. Hannah, C. Macdonald, B. He, J. Peng, and I. Ounis. University of Glasgow at TREC 2007: Experiments in Blog and Enterprise Tracks with Terrier. In *Proceedings of TREC 2007*, 2008.

[8] B. He. *Term Frequency Normalisation for Information Retrieval*. PhD thesis, Univ. of Glasgow, 2007.

[9] B. He, and I. Ounis. Combining fields for query expansion and adaptive query expansion. *Information Processing and Management* 43(5):1294–1307, 2007.

[10] A. Java, P. Kolari, T. Finin, A. Joshi, and T. Oates. Feeds That Matter: A Study of Bloglines Subscriptions. In *Proceedings of ICWSM 2007*, 2007.

[11] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[12] P. Kolari, T. Finin, A. Java, and A. Joshi. Spam in Blogs and Social Media, Tutorial . In *Proceedings of ICWSM 2007*, 2007.

[13] K. L. Kwok, and M. Chan. Improving two-stage ad-hoc retrieval for short queries In *Proceedings of SIGIR 1998*, pages 250–256, 1998.

[14] C. Lioma, C. Macdonald, V. Plachouras, J. Peng, B. He and I. Ounis. University of Glasgow at TREC 2006: Experiments in Terabyte and Enterprise Tracks with Terrier. In *Proceedings of TREC 2006*, 2007.

[15] C. Macdonald and I. Ounis. The TREC Blogs06 collection : Creating and analysing a blog test collection. Technical Report TR-2006-224, Univ. of Glasgow, 2006.

[16] C. Macdonald and I. Ounis. Voting for Candidates: Adapting data fusion techniques for an Expert Search task. In *Proceedings of CIKM 2006*, 2006.

[17] C. Macdonald and I. Ounis. Expertise Drift and Query Expansion in Expert Search. In *Proceedings of CIKM 2007*, 2007.

[18] C. Macdonald and I. Ounis. Searching for Expertise: Experiments with the Voting Model. In *Special Issue of the Computer Journal on Expertise Profiling*. 2008; doi: 10.1093/comjnl/bxm112

[19] C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC-2007 Blog Track. In *Proceedings of TREC-2007*, 2008.

[20] G. Mishne and M. de Rijke. A study of blog search. In *Proceedings of ECIR 2006*, pages 289–301, 2006.

[21] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A high performance and scalable information retrieval platform. In *Proceedings of the OSIR Workshop 2006*, pages 18–25, 2006.

[22] I. Ounis, M. de Rijke, C. Macdonald, G. Mishne, and I. Soboroff. Overview of the TREC-2006 Blog Track. In *Proceedings of TREC-2006*, 2007.

[23] C. J. van Rijsbergen. Information Retrieval, 2*ed*. Butterworths, 1979.

[24] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at TREC-2. In *Proceedings of TREC-2*, pages 21–34, 1994.

[25] S. Robertson, H. Zaragoza and M. Taylor. Simple BM25 Extension to Multiple Weighted Fields. In *Proceedings of CIKM 2004*, pages 42–49, 2004.

[26] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of SIGIR 1996*, pages 21–29, 1996.

[27] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2006.

[28] Terrier 2.1 documentation: Examples of using Terrier to index TREC collections: WT2G and Blogs06, 2008. `http://ir.dcs.gla.ac.uk/terrier/doc/trec_examples.html`.

[29] M. Thelwall. Bloggers during the London attacks: Top information sources and topics. In *Proceedings of WWW Workshop on the Weblogging Ecosystem*, 2006.