Key Blog Distillation: Ranking Aggregates

Author:Craig Macdonald,Iadh Ounis

CIKM'08

Speaker:Yi-Lin. Hsu

Advisor:Dr. Koh, Jia-Ling

Date:2009/4/27

Outline

- Introduction
- Experiment Setup
- Experiment Result
- Conclusion & Future work

Introduction

- 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.

Introduction

- 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.

Introduction

- 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

BLOG RETRIEVAL AT TREC

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.

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.*

Ranking Aggregates

- 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.

Ranking Aggregates

- 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 :

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

  - R(Q) is the underlying ranking of blog posts
  - posts(B) is the set of posts belonging to blog B

Ranking Aggregates

- 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.

Ranking Aggregates

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

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

a

Ranking Aggregates

- 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:

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

Ranking Aggregates

- the expCombMNZ technique is similar to expCombSUM, except that the count of the number of retrieved posts is also taken into account:

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

Ranking Aggregates

- the expCombMNZ technique is similar to expCombSUM, except that the count of the number of retrieved posts is also taken into account:

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

EXPERIMENTAL SETUP

- we have two forms of alternative content that can be indexed for each post
  - the XML content
  - the HTML permalinks
- the two alternative ranking strategies
  - voting techniques
  - virtual documents

EXPERIMENTAL SETUP

- A large ***virtual document*** containing all term occurrences from all of its constituent posts (either permalink content or XML content) concatenated together.
- 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.

EXPERIMENTAL SETUP

	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.

EXPERIMENTAL SETUP

- We rank index ***entities*** (whether virtual documents or posts) using the new DFRee Divergence from Randomness (DFR) weighting model. In particular, we score an entity e (i.e. a blog or a blog post) with respect to query Q as:

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

EXPERIMENTAL SETUP

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

- Prior = tf/length
- post =(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
- TFC is the number of tokens in the entire collection.

EXPERIMENTAL SETUP

- 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 . 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 social ranking of systems in TREC 2007 was done by title-only systems.

EXPERIMENTAL SETUP

```
<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.

- Retrieval performance is reported in terms of Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Precision @ rank 10 (P@10).

EXPERIMENTAL RESULTS

- In our experiments, we aim to draw conclusions on several points:
- Firstly, can indexing using only the textual con-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
- Lastly, given that we experiment with various possible voting techniques, whether there is any variance between the techniques.

EXPERIMENTAL RESULTS

	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.

CENTRAL & RECURRING INTERESTS

- **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.

CENTRAL INTERESTS

- We apply a single-pass clustering algorithm to cluster all the posts of the blogs with more than θ posts.
- 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.

CENTRAL INTERESTS

- 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.

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

CENTRAL INTERESTS

- Moreover, if no clustering has been applied for the blog (i.e. the blog has less than posts), then QscoreCluster(p,B) = 0. We integrate the clusters quality score with the exp-CombMNZ voting techniques for scoring a blog to a query

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

RECURRING INTERESTS

- We believe that a relevant blog will continue to post relevant posts throughout the timescale of the collection. We break the 11 week period 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 as follow:

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

- dateIntervali(posts(B)) is the number of posts of blog B in the ith date interval.

RECURRING INTERESTS

- We integrate the QscoreDates(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 corpustimespan is 11 weeks)

Focused Interests

- 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:

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

Focused Interests

- We integrate the cohesiveness score with the score(B,Q) for a blog to a query as follows:

$$\begin{aligned} score(B,Q) &= score(B,Q) \\ &+ log(1+\omega \cdot Cohesiveness_{Cos}(B)) \end{aligned}$$

- Where $\omega > 0$ is a free parameter.

Results and Analysis

	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$\gg$	0.7665	0.4822
+ Dates	$\omega = 0.48$	**0.2788**$\gg$	**0.7893**	**0.5022**$\gg$	$\omega = 3.49$	**0.2980**$\gg$	0.7707	**0.5289**$\gg$
+ Cohesiveness (HTML)	$\omega = 1.4$	0.1847$\ll$	0.7719	0.3556$\ll$	$\omega = 0.003$	0.2577	**0.7747**	0.4733
+ Cohesiveness (XML)	$\omega = 0.0035$	0.2280$\ll$	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 $\ll$.

CONCLUSIONS & FUTURE 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.

CONCLUSIONS & FUTURE WORK

- 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.

CONCLUSIONS & FUTURE WORK

- 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.

Future Works

- 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.