TREC 2007 Blog Track Experiments at Kobe University

Kazuhiro Seki, Yoshihiro Kino, Shohei Sato, and Kuniaki Uehara
Kobe University
1-1 Rokkodai, Nada, Kobe 657-8501, Japan

seki@cs.kobe-u.ac.jp

Abstract

This paper describes our approaches to the opinion retrieval and blog distillation tasks for the Blog Track. For opinion retrieval we employ a two-stage framework consisting of keyword search and opinion classification, where customer reviews collected from Amazon.com are used for feature selection. For the blog distillation task we consider all the blog posts belonging to a blog in order to estimate the relevance of the blog at large. To accomplish this, we first identify relevant blogs for a given topic by keyword search and then examine all the posts for each identified blog. In addition, we attempt to detect and discard spam blogs (splogs) and non-English blogs to improve system performance.

1 Introduction

In 2006, TREC introduced a new track, the *Blog Track*, which received 57 runs from 14 different research groups world-wide [5]. For the second year of the track, two tasks were tackled; one was the *opinion retrieval* task—continuing from the first year—and the other was the *blog distillation* task. We took part in both tasks, except for the *polarity* sub-task of opinion retrieval. The following reports on our approaches to the two tasks and the results of the official runs and some additional experiments.

2 Blog Distillation

2.1 Task

The blog distillation task aims at finding key blogs for a given topic that one would like to add to her RSS reader and to read on a regular basis. Note that, unlike opinion retrieval, this task is not concerned with whether the blogs are opinionated or not. There are 45 topics created in advance by the participants, and for each topic we return a ranked list of up to 100 blog feeds. The primary metric used for this task is MAP. More details can be found at the official Blog Track Wiki.[1]

2.2 Our Approach

For this task, we consider all the blog posts belonging to a blog in order to estimate the relevance of the blog at large. To accomplish this, first we search for blogs that are possibly relevant to a given topic using its title as a query. Then, after discarding non-English blogs and splogs as described shortly, we search for blog posts belonging to each of the initially retrieved blogs using the same query (topic title). In order to assess the relevance of each blog for the topic, we plot the topical relevance (cosine similarity in this study) of the blog posts for the blog along their posing dates. We regard the area under the line as the overall relevance of the blog. Figure 1 illustrates our proposed framework described above.

The following sections detail some of the key components within the framework.

2.2.1 Non-English Blog Detection

As the Blog Track targets only English contents, all non-English blogs can be seen as noise [8]. Thus, it is expected that, if one could precisely detect and discard non-English blogs, it would improve system performance. To this end, we employ a character-based *n*-gram model to distinguish non-English from English blogs.

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

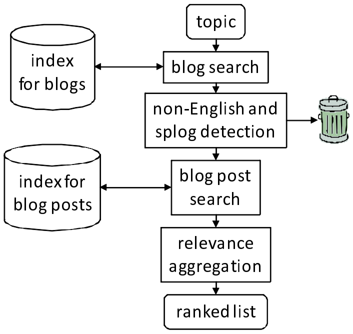

Figure 1: Our proposed framework for blog distillation.

For this experiment, we built a character-based 3-gram model with the simple Jeffreys-Perks law smoothing [4] using the Reuters-21578 corpus consisting of 21,578 news stories.[2] For each permalink file in the blog06 corpus, we extracted its textual contents (represented as a character sequence $c_1 \ldots c_n$) and computed the cross entropy on the 3-gram model as in

$$\begin{aligned} H &\approx -\frac{1}{n} \log P(c_1 \ldots c_n) \\ &\approx -\frac{1}{n} \sum_i \log P(c_i | c_{i-2}, c_{i-1}), \end{aligned}$$

with a 2nd order Markov approximation. The lower the value of H is, the more likely the input character sequence $c_1 \ldots c_n$ is English. When H for a given character sequence is greater than a predefined threshold t, we regarded it as non-English. We set $t = 3.553$ such that t separates two normal curves, each representing English and non-English contents, with the predicted error rate being 1% for English.

2.2.2 Graphical Relevance Aggregation

Given a set of initially retrieved blogs obtained by "blog search" (see Figure 1), we estimated the overall relevance of each blog B for a given topic by graphical relevance aggregation described below. We first computed the cosine similarity score s_i between the topic title q and each post $b_i \in B$ as the topical relevance of b_i. Then, those similarity scores were sorted according to the posting dates of their associated blog posts and were plotted on a graph with the x-axis being the posting dates normalized by the timespan of B. We regarded the area under the line as the overall relevance of the blog for the given topic. The rationale behind is that a good blog frequently talking about a given topic would have posts constantly having high topical relevance with the topic (query), resulting in a large area under the line. For example, Figure 2 shows topical relevance along the normalized time-line for five blogs initially retrieved for "NYC restaurants" (topic #978). Lastly, the initially retrieved blogs were ranked and output according to a descending order of the area.

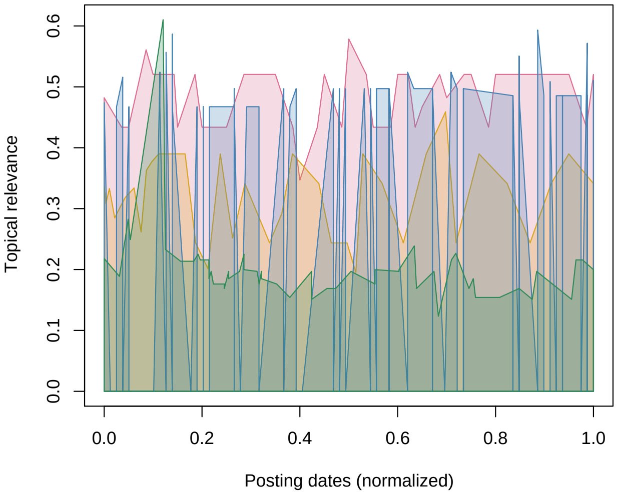

Figure 2: Example topical relevance plots of five blogs for "NYC restaurants."

2.3 Results and Discussion

We submitted four runs for this task, for which the configurations and results are summarized in Table 1. From these results, excluding splogs and non-English contents is seemingly effective for feed search; MAP improved by 5.5% compared with the baseline.

Next, to examine the effectiveness of the graphical relevance aggregation described in Section 2.2.2, we simply summed up the cosine similarity scores associated with blog posts for each blog identified by the initial blog search, and ranked the blogs in a descending order of the sums. We did not use splog and non-English contents detection for this experiment. The resulting MAP was 0.1792, indicating the validity of the graphical relevance aggregation.

An even simpler approach to perform feed search without relevance aggregation would be to output the 100 feeds that are most highly ranked by the initial blog

[2] Available at `http://www.daviddlewis.com`

Table 1: Run configurations and their performance in MAP for feed search.

Runtag	English	Splog	MAP	Description
kud	–	–	.2292	Title-only compulsory run (baseline)
kuds	–	√	.2325 (+1.4%)	Exclude splogs
kudn	√	–	.2376 (+3.6%)	Exclude non-English
kudsn	√	√	.2420 (+5.6%)	Exclude non-English & splogs

search. We tested this approach with/without splog and non-English contents detection. Table 2 shows the results in MAP and precision at 5 (P@5), where all the official runs are included for comparison.

Contrary to our expectation, it turned out that the initial-search-only run without relevance aggregation and splogs/non-English blogs detection (the top row) performed well, obtaining over 20% higher MAP than our official runs. Although we observe some improvement among the highly ranked permalinks as indicated by P@5 for some cases, only run that beaten the initial-search-only run in MAP is the one with non-English blog exclusion. In other words, splog detection and relevance aggregation did not improve MAP in the current settings.

To investigate the problem, we closely looked at the blog06 corpus and found that many permalink URLs were not properly extracted from the corresponding feed files. For example, `BLOG06-feed-000017` is associated with no permalinks in `20051206/feeds-000.gz` according to `<PERMALINKS>` tags, but the feed actually contains several permalinks, such as `Http://www.MacHall.Com?strip_id=357`. The same problem was found for `BLOG06-feed-000036`, `BLOG06-feed-000043`, and many others. Another problem is, although less frequent, that the extracted URLs are sometimes not permalinks but hyperlinks to the web pages the blog posts are commenting on. For example, see `BLOG06-feed-000065`, `BLOG06-feed-001152`, etc. These flaws may be in part harming our approach focusing on individual permalinks' topical relevance.

3 Opinion Retrieval

3.1 Task

The opinion retrieval task intends to find blog posts which do not simply mention a given topic but explicitly express opinions of the authors or commenters about it. The topic (or target) can be any concrete/abstract object including a person, a product, an event, etc.

Given 50 topics selected from a commercial blog search engine query logs, participants are required to return an ordered list of up to 1000 blog posts (documents) for each topic. The evaluation is done through the standard pooling procedure, and system performance is measured primarily by MAP.

3.2 Our Approach

Figure 3 depicts our approach to the opinion retrieval task. The following details major components of the approach.

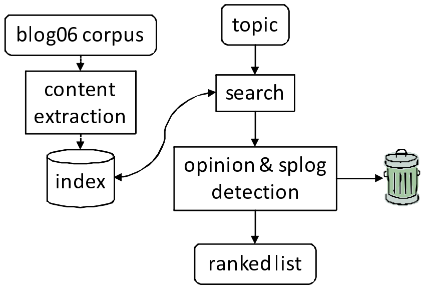

Figure 3: An overview of our approach to opinion retrieval.

Before creating an index of the blog06 corpus, we extract textual information from the permalink files. Ideally, only the contents and comments of blog posts should be extracted as permalinks contain a variety of information not necessarily related to the posts themselves, such as advertisements. For this purpose, we developed a heuristic algorithm taking advantage of feed files and HTML tag structure similarly to Liao et al. [2]. However, as our preliminary experiments showed that using only the extracted contents and comments actually degraded the system performance, the experiments

Table 2: Additional experiments for blog distillation.

Runtag	English	Splog	Rel. Agg.	MAP	P@5
–	–	–	–	.3064	.4444
–	–	√	–	.2776 (−9.4%)	.5333 (+20.0%)
–	√	–	–	.3260 (+6.4%)	.5156 (+16.0%)
–	√	√	–	.2873 (−6.2%)	.5867 (+32.0%)
`kud`	–	–	√	.2292 (−25.2%)	.2844 (−36.0%)
`kuds`	–	√	√	.2325 (−24.1%)	.4800 (+8.0%)
`kudn`	√	–	√	.2376 (−22.5%)	.3156 (−29.0%)
`kudsn`	√	√	√	.2420 (−21.0%)	.5111 (+15.0%)

reported in this paper relied on all the textual information found in permalink files. Even though it was not utilized to produce official runs, Figure 4 presents (a digest of) the extraction algorithm for completeness.

```
Input: A permalink file
Output: Post contents and comments (if any)

Construct an HTML tag tree for an input;
For each child node of <body> do
    Remove tags and their contents which are
    normally not related to post contents (e.g.,
    <script> and <style>);
    Remove tags which do not bear textual contents (e.g., <br> and <hr>);
    Remove tags with irrelevant id or class values
    (e.g., advertisement) and their contents;
    Remove tags with five or more <a> tags as
    their children;
end
Locate and extract post contents based on the post title
and snippet obtained from the corresponding RSS;
Locate and extract comments based on a <form> tag;
```

Figure 4: An algorithm of contents and comment extraction from permalink files. (Not used for official runs.)

For indexing and retrieval, we used an open source IR system, Apache Lucene,[3] equipped with the vector space model [6] and the tf-idf term weighting scheme [7]. Stopwords were removed and all terms were lower-cased in indexing. No stemmer was applied.

Given a topic, we utilized `<title>` fields only as a query. In this initial retrieval phase, we limited the number of retrieved documents (permalinks) to the top 2000. Then, opinion and splog classifiers were independently applied to those 2000 documents, where Support Vector Machines (SVMs) [1] and k-nearest neighbors (kNN) were experimentally tested. Instead of discarding blogs marked as "splog" or "non-opinion," we integrated classification results (confidence for prediction) with the retrieval results (cosine similarity) in the form of their product.

As the training data set for learning the opinion classifier, we used the qrels from the 2006 Blog Track where 11,234 and 8,280 documents are manually labeled as opinions (labels 2–4) and non-opinions (label 1), respectively. Note, however, that we explored to use 27,544 positive/negative customer reviews harvested from `Amazon.com` in order to find good sentiment terms as features.

For splog classification, 10,000 splogs randomly selected from SplogSpot[4] were used as positive instances. On the other hand, since we do not have a good set of non-splogs, we regarded the relevant permalinks—whether opinionated or non-opinionated—from the Blog Track 2006 qrels as non-splogs, considering the fact that permalinks judged as relevant are typically non-splogs. To be precise, only 8.3% of relevant permalinks are reported to be splogs [5].

As with the case of the opinion classifier, individual terms—which Lin et al. [3] reported to be quite effective—were used as features for splog classification. Additionally, we used two features based on hyperlink information. One was the variance of outlink URL frequencies within a permalink, which intended to capture the characteristic that splogs tend to have a number of links pointing to the same URL. The other link-based feature was the variance of the number of

[3] `http://lucene.apache.org/`

[4] `http://www.splogspot.com/`

Table 3: Run configurations and their performance in MAP for opinion retrieval.

Runtag	Opinion	Splog	MAP	Description
Ku	–	–	.1689	Title-only compulsory run (*baseline*)
KuKnn	kNN	–	.1657 (-1.9%)	Use kNN for opinion classification
KuSVM	SVM	–	.1644 (-2.7%)	Use SVM for opinion classification
KuS	–	SVM	.1580 (-6.5%)	Use SVM for splog classification
KuSVMS	SVM	SVM	.1555 (-7.9%)	Use SVM for opinion and splog classification

unique anchor texts pointing to the same URL, which represented our observation that splogs often use different anchor texts to link to the same website.

3.3 Results

We submitted five runs for the opinion retrieval task. The configurations of those runs and their performance in MAP for opinion retrieval (not mere topical relevance) are presented in Table 3.

In brief, our attempts to separate opinions from non-opinions and to separate splogs from non-splogs all failed. We are currently inspecting the results to identify what has caused the problems.

4 Concluding Remarks

This year, we participated in the Blog Track and tackled the blog distillation and opinion retrieval tasks. Unfortunately, the empirical results indicate only limited success with respect to our attempts to exclude noise (i.e., splogs and non-English contents) from the system output and to estimate the overall blog relevance. We are currently analyzing the results and are examining our approaches for improvement.

References

[1] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML 98)*, pages 137–142, 1998.

[2] Xiangwen Liao, Donglin Cao, Songbo Tan, Yue Liu, Guodong Ding, and Xueqi Cheng. Combining language model with sentiment analysis for opinion. In *Proceesings of the 15th Text Retrieval Conference*, 2006.

[3] Yu-Ru Lin, Wen-Yen Chen, Xiaolin Shi, Richard Sia, Xiaodan Song, Yun Chi, Koji Hino, Hari Sundaram, Jun Tatemura, and Belle Tseng. The splog detection task and a solution based on temporal and link properties. In *Proceesings of the 15th Text Retrieval Conference*, 2006.

[4] Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

[5] Iadh Ounis, Maarten de Rijke, Craig Macdonald, Gilad Mishne, and Ian Soboroff. Overview of the TREC-2006 blog track. In *Proceesings of the 15th Text Retrieval Conference*, 2006.

[6] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1983.

[7] Karen Sparck Jones. Statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–20, 1972.

[8] Kiduk Yang, Ning Yu, Alejandro Valerio, and Hui Zhang. WIDIT in trec-2006 blog track. In *Proceesings of the 15th Text Retrieval Conference*, 2006.