Multiple Ranking Strategies for Opinion Retrieval in Blogs

The University of Amsterdam at the 2006 TREC Blog Track

Gilad Mishne
ISLA, University of Amsterdam
gilad@science.uva.nl

Abstract

We describe our participation in the Opinion Retrieval task at TREC 2006. Our approach to identifying opinions in blog post consisted of scoring the posts separately on various aspects associated with an expression of opinion about a topic, including shallow sentiment analysis, spam detection, and link-based authority estimation. The separate approaches were combined into a single ranking, yielding significant improvement over a content-only baseline.

Introduction

The task in the Blog Track introduced in TREC this year was *opinion retrieval*: identifying and ranking blog posts expressing an opinion regarding a given topic. Our approach to this task was to identify different components which indicate such an opinionated expression, rank the blog posts according to each of these separately, and combine these partial relevance scores to a final one. This allows us to break down the opinion retrieval task to a number of simpler subproblems, which we treat as independent.

We proceed by describing the components of opinionated relevance we identified, how a score was calculated for each, and how the final score was derived.

Opinion Retrieval Components

We identify three different aspects indicating that a blog post expresses an opinion about a topic: *topical relevance*, *opinion expression*, and *post quality*. The first aspect, topical relevance, is the degree to which the post deals with the given topic; this is similar to relevance as defined for ad-hoc retrieval tasks such as many of the traditional TREC tasks. The second aspect, opinion expression, refers to identifying, given a "topically-relevant" blog post, whether it contains an opinion about topic: to what degree it contains subjective information about it. Finally, post quality is an estimation of the (query-independent) quality of a blog post, under the assumption that higher-quality posts are more likely to contain meaningful opinions and are preferred by users.

Note that a relevant blog post, as defined in the opinion retrieval track, does not necessarily have high topical relevance, or high post quality: a document is relevant if it contains an opinion about the target, even if the target is not the main topic of the document and the opinion is expressed only in passing. However, cursory examination of posts containing opinions about various targets shows that in the majority of the cases, the target is also the main topic; an in-depth analysis to examine the degree to which this assumption holds is planned for future work.

Topical relevance

To estimate the ad-hoc relevance score of a blog post given a topic, we used a straightforward retrieval approach, enhanced by a few heuristics. As the basic relevance score, we use a language modeling based retrieval method shown to achieve same-or-better scores when compared to top-performing retrieval algorithms (5). The TREC Blogs06 corpus contains separate collections of the post feeds, permalink HTML pages, and blog home pages (9). For our experiments, we used the text appearing in the feed part of the collection, except where the feed was a partial content one – in which case, the text was extracted from the appropriate HTML page (similar to (4)). The blog home pages were not used. Standard tokenization and English Snowball stemming were applied; anchor text was extracted and added to the text of the linked post.

We experimented with a number of simple techniques for improving the retrieval achieved with plain language modeling ranking. The first is blind relevance feedback in the language modeling framework as proposed by (15). Essentially, this method adds terms to the original query by comparing the language model of the top-retrieved documents with the model of the entire collection, adding terms which are indicative of these top-retrieved documents. As with other query expansion methods, this type of relevance feedback is known to increase recall at the expense of precision; since topical relevance is only one component in our system, we decided in favor of using it, hypothesising that other retrieval components will balance the precision drop. However, to prevent exsessive topic drift, we limited the number of added terms to 3; examples of terms added to real topics are shown in Table 1.

Topic	Added terms
859. letting india into the club?	nuclear, times, friedman
867. cheney hunting	vice, dick, accident
896. global warming	climate, change, greenhouse

Table 1: Examples of terms added with relevance feedback

The next heuristic we applied to improve topical relevance was usage of term proximity. Recent work has shown that, contrary to previous results, taking proximity into account improves retrieval substantially (10), in particular in web-type documents such as the ones in our collection (11). In particular, we used the method described in (11) where every word n-gram ($n > 1$) of the topic is used as a query term, boosting the score of documents which match phrases appearing in the topic.

The final technique we employed in the framework of topical relevance was usage of the temporal properties of the collection. According to a study of a blog search engine log (12), a substantial amount of blog queries are *recency queries* – queries which favor recent documents, rather than having an even distribution of relevance. Since the blogspace is a highly dynamic domain, many queries are related to current events – possibly, to evaluate how bloggers react to these events. Consequently, it seems useful to assign a higher relevance to blog posts which were "recent" at the time the query was issued, as described in (8). However, as TREC topics are not distributed with an accompanying query issue time, we use the following method to estimate it. First, we use a plain topical relevance model to retrieve the top 100 blog posts which are relevant for a query. Since every post in the corpus has an associated timestamp, we can now observe the distribution of dates in these top-100 posts. We adopt a simple approach and assume that the query was issued in the top-occurring date.[1] Table 2 shows some "query issue dates" derived using this approach, with an accompanying post-analysis.

Once we have the estimated query issue date, we use a linear combination of a document's retrieval rank with its recency rank to boost the scores of posts published close to the time of the query date. This method has been described in (8), and shown to improve retrieval performance substantially for recency queries. Note that not all queries are indeed recency ones – for example, the last query in Table 2 is not necessarily time-related; identifying these queries is relatively simple, as the distribution of dates in the top results does not contain peaks. For the experiments reported here, we treated all queries as recency ones, and intend to address the identification of the non-recency ones in future work.

Opinion expression

Sentiment analysis is an area of research dealing with the extraction and characterization of emotions, opinions, and other non-factive aspects of text; work in this domain in recent years is plentiful (e.g., (17)). Within this area, a key task is *sentiment classification* – identifying positive and negative opinions towards a given topic; this task has also been previously investigated at the Novelty track at TREC (18).

Broadly speaking, there are two approaches to sentiment classification: lexicon-based methods and machine learning approaches. Lexical method first construct a dictionary of terms indicating sentiment (often, lists of "positive" and "negative" words); sometimes, words are associated with weights. The sentiment of a given text is then derived by the occurrence of words from this dictionary in the text, e.g., by summing their weights. There are various ways of building the sentiment dictionary, including pattern-based (16), using WordNet (6), co-occurrence statistics (20) and more. Machine learning methods train a classifier using a set of annotated texts containing sentiment, typically employing features such as n-grams of words, part-of-speech tags, and logical forms (3, 14).

We view the ranking of blog posts for opinion expression as a sentiment classification task, and approach it with a lexicon-based method. The lexicon we use is the General Inquirer (19), a large-scale, manually-constructed lexicon assigning a wide range of categories to more than 10,000 English words. Among the categories assigned are Osgood's semantic dimensions and emotional categories. The following word categories are used as indicators of the existence of an opinion in the text: the two valence categories, *Positive* and *Negative*; the emotional categories, *Pleasure*, *Pain*, *Feel*, *Arousal*, *EMOT*, *Virtue*, and *Vice*; the pronoun categories, *Self*, *Our*, and *You*; the adjective categories, *IPadj* (relational adjectives) and *IndAdj* (independent adjectives); and the *Respect* category.[2]

For each post, we calculate two sentiment-related values using the words appearing in each of these categories: a "post opinion level" and a "feed opinion level." In both cases, the opinion level is the number of occurrences of words from any of these categories, normalized by the total number of words: the difference is the text used for counting the occurrences. For the post opinion level, we extract all "topical sentences" from the post, using them as the text to count the opinion-bearing words in. Topical sentences, in our approach, are all sentences containing the topic verbatim, as well as the sentences immediately surrounding them. This is done to focus the search for opinion-bearing words to parts of the post which are likely to refer directly to the topic, rather than the post in its entirety. For the second value, the feed opinion level, we use the entire feed to which the post belongs; this is a static, topic-independent score per feed, estimating the degree to which it contains opinions (about any topic). The intuition here is that feeds containing a fair amount of opinions are more likely to express an opinion in any of their given posts; since the amount of text in a feed is typically substantially larger than that of a single post, and since lexical methods such as the one we use work better on longer texts, this may yield a more robust measurement.

Post quality

Finally, the third set of rankings we use for determining relevance for the opinion retrieval task concerns the "quality" of the blog post. In particular, we are interested in filtering spam posts, and in incorporating the degree of authority assigned to a post (and its feed) into the final retrieval score. While posts containing opinions do not necessarily have high authority, we hypothesized that, given two posts

[1] In practice, we employ some additional heuristics here, such as working with windows of a few days rather than a single day – mostly to account for differences in time zones and so on.

[2] A complete list of the General Inquirer categories is given at http://www.wjh.harvard.edu/∼inquirer/homecat.htm

Topic	Estimated date	Explantion
853. state of the union	01-Feb-2006	President Bush delivers the State of the Union Address on the evening of January 31st, 2006.
874. coretta scott king	08-Feb-2006	Coretta Scott King's funeral was held on February 8th, 2006.
891. intel	11-Jan-2006	Apple announces first Intel-based computers on January 10th, 2006.

Table 2: Examples of estimated "query issue dates"

with similar opinions, a searcher will prefer the one with higher authority. For this, we calculate separate spam and authority scores, both of which are query-independent.

Link-based Authority. Estimating the authority of documents in a hyperlinked environment using an analysis of the link structure is known to be an effective approach (e.g., PageRank, HITS). We follow Upstill et al. (21), which show that the inbound link degree is a good approximation of more complex approaches such as PageRank. To this end, we use both the log of the inbound link degree of a post p (discarding links from other posts which belong to the same feed as p) and the inbound link degree of p's feed (again, discarding intra-feed links) as a crude estimation of the post's authority.

Spam Likelihood. Spam blogs are an increasing nuisance in the blogspace (7), and the TREC collection is no exception. Some spamming techniques result in high topical retrieval scores for certain queries on spam blog posts; to address this, we employed a relatively simple spam filtering mechanism, in which a "spam likelihood" score was assigned to each feed using two independent methods.

First, we used a machine-learning approach which has been shown to be effective for spam detection in this domain (7). We created a training set of spam and non-spam feeds using two naive assumptions: The first assumption is that a feed from the domain `blogspot.com`, and with a domain name exceeding 35 characters, is a spam blog. Sample domains which are judged as spam using this rule are `casino-hotel-in-windsor-poker.blogspot.com` or `weightloss7666resources.blogspot.com`. The second naive assumption is that a feed from the domains `livejournal.com` or `typepad.com` is not spam. While both assumptions are crude, we found that they achieve very high precision (at the expense of low recall).[3] Our training collection was created by randomly sampling 500 feeds which meet the "spam assumption", and 500 feeds which meet the "non-spam" one; this provided us with a relatively high-quality, if biased, collection. We then trained an SVM on this set, and used its prediction scores on the entire collection as one evidence for the likelihood of a given feed to be spam.

Our second spam detection method follows one of the techniques used by Ntoulas et. al (13), namely, text-level compressibility. Many spam blogs use keyword stuffing – a high concentration of certain words, aimed at obtaining high relevance scores from search engines for these keywords. Keywords and phrases are often repeated dozens and hundreds of times in the same blog "post", and across posts in the spammy feed; this results in very high compression ratios for these feeds, much higher than those obtained with non-spam feeds. Using the same collection as used for training the SVM, we calculated the distribution of compression ratios of both spam and non-spam feeds (which differed substantially); all feeds in the corpus were then assigned a likelihood of being drawn from each of these distributions.

The final spam likelihood estimate is a product of the SVM prediction and the compressibility prediction.

Model Combination

Combining retrieval scores assigned by different methods to the same set of documents is a common task in IR, particularly in web domains (see, e.g., (2)). We adopt one of the standard methods used in this scenario – computing the final retrieval score as a linear combination of the various partial scores. More concretely, the score is calculated as follows. First, we retrieve the top-1000 posts using topical relevance, language-modeling retrieval only. These 1000 posts are then scored also using the other methods we described (such as link indegree, post opinion level, and so on). Each of these methods, as well as the plain language-modeling similarity, has a (static) associated weight; the final score is a linear combination of the scores assigned by the methods, multiplied by their respective weights.

Model Weighting. Clearly, the performance of such a linear combination approach depends to a large extent on the weight assigned to each component. Optimizing the weights requires training data; but, as this is the first run of the opinion retrieval task at TREC, such data was not available. We therefore decided to create partial relevance scores by assessing a limited number of the top-ranking documents retrieved using a small set of 10 training topics we developed.[4] We retrieved the top-50 posts for each of these topics, using plain language-modeling ranking, and judged their relevance according to the assessor guidelines used at TREC. Weights of the linear combination were then optimized, where the value to maximize was the bpref score of the combined ranking; bpref was used as it was shown to be more stable to partial judgment scores (1) – which is most probably the case we are facing, given our limited assessment effort. The final weights used are shown in Table 3.

Submissions and Results

We submitted 5 runs, as follows:

[3] Our assumptions are based on the popularity of Blogspot among spammers due to easy automation of posting at the time of collecting the Blogs06 corpus, and a relatively low level of spam in TypePad (a platform requiring payment) and LiveJournal.

[4] The topics were "Windows Vista," "Hurricane Katrina," "Oscar," "Pepsi," "oil prices," "iPod Nano," "EU Constitution," "Katie Holmes," "Lebanon," and " Paris Riots."

Component	Weight
Content-based retrieval (with query expansion)	1.00
Recency score	0.03
Post opinion level	0.10
Feed opinion level	0.45
Link-based authority	0.01
Spam likelihood	0.90
Phrase matching score	0.05

Table 3: Component weights

- `UAmsB06Base`: A baseline run, consisting of language modeling based retrieval using the terms appearing in the *title* field of the query, with blind relevance feedback as described in Section .
- `UAmsB06L`: Same as `UAmsB06Base`, with link-indegree reranking.
- `UAmsB06S`: Same as `UAmsB06Base`, with spam-detection.
- `UAmsB06O`: Same as `UAmsB06Base`, with opinion reranking – both at the feed level and the post level.
- `UAmsB06All`: All components used: link indegrees, spam detection, opinion expression (feed and post), time models, and phrase-level reranking.

The retrieval scores for these runs are shown in Table 4.

Run	MAP	R-Prec	bpref
`UAmsB06Base`	0.1449	0.2357	0.2393
`UAmsB06L`	0.1417	0.2269	0.2375
`UAmsB06S`	0.1523	0.2448	0.2485
`UAmsB06O`	0.1596	0.2573	0.2509
`UAmsB06All`	**0.1795**	**0.2771**	**0.2625**

Table 4: Scores of submitted runs

The assumption that posts with higher link-authority will be preferred by users turned out to be incorrect: examining the qrels, the average link indegree of relevant posts (2.4) was lower than that of non-relevant ones (4.2); indeed usage of link-based authority scores decreased the accuracy of the baseline. All other methods we used improved the baseline to varying extents, and the combination of all methods (optimized for our small training set) yielded an improvement of 24% to MAP and 18% to R-Precision. Of all components, the one contributing most to the improvement was the shallow sentiment-based opinion expression module.

Conclusions

We presented our approach to opinion retrieval in blogs, consisting of combining topical retrieval with a number of aspects we view as contributing to relevancy for this task, including shallow sentiment analysis, spam detection, and link-based authority. While some approaches, most notably the sentiment analysis, contributed to significant improvements over topical retrieval alone, other approaches such as link-based ones degraded performance. Overall, the combination of all approaches substantially improved topical-relevance retrieval only. In future work, we intend to examine more closely the contribution of each component to accuracy, as well as explore different combinations of the components.

Acknowledgments. Thanks to Breyten Ernsting for help in processing the collection. This work was supported by the Netherlands Organization for Scientific Research (NWO) under project number 220-80-001.

References

[1] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04*, 2004.

[2] N. Craswell, S. Robertson, H. Zaragoza, and M. Taylor. Relevance weighting for query independent evidence. In *SIGIR '05*, 2005.

[3] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW '03*, 2003.

[4] N. Glance. Indexing Weblogs One Post at a Time. In *AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*, 2006.

[5] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, Enschede, Jan. 2001.

[6] S.-M. Kim and E. Hovy. Automatic detection of opinion bearing words and sentences. In *IJCNLP-05*, 2005.

[7] P. Kolari, T. Finin, and A. Joshi. SVMs for the Blogosphere: Blog Identification and Splog Detection. In *AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*, 2006.

[8] X. Li and W. B. Croft. Time-based language models. In *CIKM '03*, 2003.

[9] C. Macdonald and I. Ounis. The trec blogs06 collection: Creating and analysing a blog test collection. Technical Report TR-2006-224, 2006.

[10] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *SIGIR '05*, 2005.

[11] G. Mishne and M. de Rijke. Boosting Web Retrieval through Query Operations. In *ECIR 2005*, 2005.

[12] G. Mishne and M. de Rijke. A study of blog search. In *ECIR 2006*, 2006.

[13] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *WWW 2006*, 2006.

[14] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *EMNLP 2002*, 2002.

[15] J. M. Ponte. Language models for relevance feedback. In *Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval*, 2000.

[16] E. Riloff, J. Wiebe, and T. Wilson. Learning subjective nouns using extraction pattern bootstrapping. In *CoNLL-2003*, 2003.

[17] J. G. Shanahan, Y. Qu, and J. Wiebe. *Computing Attitude and Affect in Text: Theory and Applications (The Information Retrieval Series)*. Springer, 2005.

[18] I. Soboroff and D. Harman. Overview of the trec 2003 novelty track. In *TREC*, 2003.

[19] P. Stone, D. Dunphy, M. Smith, and D. Ogilvie. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge, 1966.

[20] P. D. Turney and M. L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.*, 21(4), 2003.

[21] T. Upstill, N. Craswell, and D. Hawking. Predicting fame and fortune: Pagerank or indegree? In *ADCS2003*, 2003.