WHU at Blog Track 2007

Haozhen Zhao，Zhicheng Luo, Wei Lu
Center for Studies of Information Resources, School of Information Management
Wuhan University, China and City University
{ zhaohaozhen@gmail.com, luozhicheng.dut@gmail.com,reedwhu@gmail.com}

Abstract

We participate in all the sub tracks of the Blog track 2007. For the Opinionated Task, we use an excerpt list of the SentiWordNet to determine the opinionated nature of returned blog posts. For the Topic distillation Task, we test the effectiveness of cleaning work for the data collection in improvement of retrieval performance and the use of title and description part as queries. Both results show that our method does not work well.

1 Introduction

This is our first time to participate in the TREC Blog Track. Our main purpose this year is to implement the blog track on our OKAPI system, and to test our roughly proposed ideas for the opinionated task and topic distillation task.

We participate in all the sub tracks of the Blog track 2007. For the Opinionated Task, we use the topic titles and an excerpt list of the SentiWordNet[1] as two groups of queries to retrieve the permanlinks part of the BLOGS06 data collection, then rank the results in two ways. One is based on the product of docweight-for-topic and the sum of docweight-for-Sentiwords, the other the docweight-for-topic after ruling out those that do no reach a certain threshold value for the docweight-for-Sentiwords. Not only do both have very dismal results, but also there is almost no difference between them.

For the Topic distillation Task, we test the effectiveness of cleaning work for the data collection in improvement of retrieval performance and the use of title and description part as queries. The results show that our method has a rather dismal outcome on both sides and using descriptions as queries outperformed both over raw data as well as cleaned data.

2 Retrieval Model

In our experiments, we use the BM25 as the core retrieval model. BM25 is a series of probabilistic models derived by Robertson et al [2] for document level retrieval. The formula used in our experiment is as follows:

$$w_j(\overline{d},C) = \frac{(k_1+1)tf_j}{k_1((1-b)+b\frac{dl}{avdl})+tf_j}\log\frac{N-df_j+0.5}{df_j+0.5} \qquad (1)$$

, where *C* denotes the document collection, tf_j is the term frequency of the *j*th term in document *d*, df_j is the document frequency of term *j*, *dl* is the document length, *avdl* is the average document length across the collection, and k_1 and *b* are tuning parameters which normalize the term frequency and element length.

Then the document score is obtained by term weights of terms matching the query q:

$$w(d,q,c) = \sum_j w_j(d,c)\cdot q_j \qquad (2)$$

3 Opionion retrieval task

3.1 Model for Opinion retrieval task

The aim of opinion retrieval task is to find opinionated blog posts on a given topic. We view the solution to this problem in two steps: one is to locate blog posts relevant to the topic, then judge its opinionated nature. Our experimental steps are as follows:

Indexing the collection We use the OKAPI system to index the permanlinks part of the collection.

Opinionated nature detection We choose the SentiWordNet, a list cosisting 115,341 words marked with positive and negative orientation scores ranging from 0 to 1. We extracted a subset of 7999 opinionated words from the SentiWordNet according to the principle that the selected words' orientation strength is over 0.4, which is similar to Attardi et al.'s work [3].

Search Strategy and Ranking We use the 50 topic titles and the subset of SentiWordNet to retrieve the indexed collection. As a result, we get all the documents relevant to a given topic and all the relevant documents to a given opinionated word in the excerpt SentiWordNet list. Now we have three tables of data, as follows.

For each word in our subset of SentiWordNet list, we have Table 1. In which PosScore represents the word's positive orientation strength while NegScore represents the negative one. SentiScore is the sum of the absolute value of PosScore and NegScore.

Table 1 SentiWordNet list

Word NO.	Word	PosScore	NegScore

For each topic, we have Table 2. In which DocNo represent the No. of the relevant document to the topic and DocWeight represents its weight.

Table 2 Relevant Documents for Topics I

Topic NO.	Topic Title	DocNo.	DocWeight

For each word in our subset of SentiWordNet list as queries to the indexed collection, we have Table 3. In which DocNo represent the No. of the relevant document to the SentiWord and DocWeight represents its weight.

Table 3 Relevant Documents for SentiWords

SentiWord NO.	SentiWord	DocNo	SentiDocWeight

Then for each relevant document for a given topic, we have R(d) as the relevance value of the document.

$$R(d) = DocWeight \cdot \sum_i SentiDocWeight_i \cdot (|PosScore_i| + |NegScore_i|) \qquad (3)$$

, where $SentiDocWeight_i$ is the DocWeight for the ith SentiWord that is in the Document relevant to the topic.

3.2 Runs

We made three automatic runs for the opinion retrieval task.

NOOPWHU1 is for opinion task with all opinion-finding features turning off. It is based on OKAPI system.

OTWHU101 ranked based on the multiple of doc weight for Topics and Opinionated weight of each doc. It is based on Okapi system with stemmed Index.

OTWHU102 is ranked based on the doc weights for each Topic, posprocessed according to a certain threshold value of opinionated nature. It is based on Okapi system with stemmed Index.

Table 4 Results for Opinion Retrieval task.

RunID		MAP	R-prec	bpref	recip_rank	P10
NOOPWHU1	opinion	0.0011	0.0071	0.0072	0.0303	0.008
	topicrel	0.0016	0.0111	0.01	0.0539	0.02
OTWHU101	opinion	0.0011	0.007	0.0061	0.0572	0.012
	topicrel	0.0016	0.0093	0.0094	0.0786	0.022
OTWHU102	opinion	0.0011	0.0071	0.0072	0.0303	0.008
	topicrel	0.0016	0.0111	0.01	0.0539	0.02

3.3 Polarity Subtask

As for the polarity subtask, which requires us to determine whether the opinions in the retrieved blog posts are positive or negative, we solve this by assign each relevant document for a given topic a P(d) value, where

$$P(d) = \sum_i SentiDocWeight_i \cdot PosScore_i - \sum_j SentiDocWeight_j \cdot NegScore_j \qquad (3)$$

, and *i* is the number of positive words in the relevance document, *j* the number of negative words. If $P(d) \geq 0.4$, we assign the document a positively opinionated label. If $P(d) \leq 0.2$, we assign the document a negatively opinionated label. If $0.2 \leq P(d) \leq 0.4$, we assign the document a mixed polarity label.

We have two groups of result corresponding to the Opinion retrieval task.

Table 5 Results for polarity subtask

RunID	Raccuracy	Acc5	Acc10	Acc15
OTPSWHU101	0.0022	0.008	0.004	0.0027
OTPSWHU102	0.0032	0	0.004	0.0053

4 Blog distillation task

4.1 Model for Blog distillation task

We first scan the permalinks part of the collection to result a mapping between the FeedNos and the PermalinksNos or the DocNos. We use the topic as queries to retrieve the collection, result a table as blow.

Table 6 Relevant Documents for Topics II

Topic NO.	Topic Title	DocNo.	DocWeight

Then for each topic, we sum up all the DocWeights for the all the DocNo that belongs to the same FeedNo.Then we rank the FeedNos according to the value of R(d) to determine the Topical relevance of a certain feed.

$$R(d) = \sum_i DocWeight_i \qquad (4)$$

4.2 Runs

We made four automatic runs for the Blog Distillation Task.

TDWHU100 is based on OKAPI system, with the raw permanlinks part of the BLOGS06 Collection, using only title as queries. Results are ranked based on combinations of doc weights for all relevant feeds.

TDWHU101 is based on OKAPI system, with the raw permanlinks part of the BLOGS06 Collection, using the description as queries. Results are ranked based on combinations of doc weights for all relevant feeds.

TDWHU200 is based on OKAPI system, with cleaned permanlinks part of the BLOGS06 Collection, using titles as queries. Results are ranked based on combinations of doc weights for all relevant feeds.

TDWHU201 is based on OKAPI system, with cleaned permanlinks part of the BLOGS06 Collection, using the description as queries. Results are ranked based on combinations of doc weights for all relevant feeds.

Table 7

RunID	MAP	R-prec	bpref	recip_rank	P10
TDWHU100	0.0117	0.0333	0.0241	0.1315	0.0422
TDWHU101	0.0071	0.0138	0.0111	0.0534	0.0156
TDWHU200	0.0135	0.0419	0.0297	0.1386	0.0578
TDWHU201	0.008	0.0199	0.0138	0.0559	0.0156

5 Discussion and Conclusion

These frustrating results suggest that we may have some fundamental fallacies in our experiments, which we have not found out. This is our first time to deal with such a huge data collection. Perhaps we have to improve our models of retrieval to make it more adapted to the large-size web data collections.

We will further our research in the Blog track with an emphasis on its web search characteristics and try more sophisticated techniques on solving the opinionated task.

References

[1] A. Esuli, F. Sebastiani. 2005. Determining the semantic orientation of terms through gloss classification. CIKM 2005: 617-624.

[2]Stephen Robertson, Steve Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In W B Croft and C J van Rijsbergen,editors, SIGIR ’ 94: Proceedings of the 17th Annual International ACM SIGIR Conferenceon Research and Development in Information Retrieval. Springer-Verlag, 1994.

[3] Giuseppe Attardi1, Maria Simi. Blog Mining through Opinionated Words. In Proceedings of the Fifteenth Text Retrieval onference (TREC-2006), 2007