A Hybrid Method for Opinion Finding Task (KUNLP at TREC 2008 Blog Track)

Linh Hoang, Seung-Wook Lee, Gumwon Hong, Joo-Young Lee, Hae-Chang Rim
Natural Language Processing Lab., Korea University
(linh,swlee,gwhong,jylee,rim)@nlp.korea.ac.kr

Abstract

This paper presents an approach for the Opinion Finding task at TREC 2008 Blog Track. For the Ad-hoc Retrieval subtask, we adopt language model to retrieve relevant documents. For the Opinion Retrieval subtask, we propose a hybrid model of lexicon-based approach and machine learning approach for estimating and ranking the opinionated documents. For the Polarized Opinion Retrieval subtask, we employ machine learning for predicting the polarity and linear combination technique for ranking polar documents. The hybrid model which utilize both lexicon-based approach and machine learning approach to predict and rank opinionated documents are the focuses of our participation this year. Regarding the hybrid method for opinion retrieval subtask, our submitted runs yield 15% improvement over baseline.

1 Introduction

TREC 2008 Blog Track defines two main tasks: the Opinion Finding and the Blog Distillation. We participated in the Opinion Finding task which is split into three separate subtasks (our system structure for the retrieval subtasks is shown in Figure 1).

The first subtask, Ad-hoc Retrieval, involves finding blog posts which contain relevant information about a given topic. To be considered as baseline, this subtask is supposed to turn off all opinion finding features. In order to concentrate on the opinion finding methods, we simply adopted out-of-the-box models supported by Lemur toolkit[1] for this subtask. According to our empirical runs, language model turned out to be relatively better than vector space model for Ad-hoc Retrieval. We also applied several techniques for query expansion but neither proposed cluster-based expanding nor classic pseudo-relevance feedback did not help to improve the retrieval performance.

The second subtask, Opinion Retrieval, involves locating blog posts that express an opinion about a given topic. The relevant blog post must have an opinion presented in the post or in one of its comments. To deal with this subtask, we structured our approach as a two-step process, including detecting opinionated documents and ranking them. To detect the opinionated blog posts, we employed both lexicon-based approach (LE) and machine learning approach (ML). The opinion scores estimated by LE and ML are then linearly combined with topic relevance score to produce the final ranking for detected documents. For this two-step process, we proposed a hybrid method utilizing LE and ML at both the detecting step and the ranking step. In addition, we applied spam filtering as a post-processing step of the Opinion Retrieval subtask, that conducted a marginal improvement of retrieval performance.

The Polarized Opinion Retrieval subtask involves not only locating positive (resp. negative) opinionated blog posts but also ranking them according to the degree of polarity. Ranking is a new requirement for this subtask in 2008. To deal with requirement of excluding mixed documents (i.e, ones contain both positive and negative opinions), we trained a ternary classifier to classify opinionated documents into positive, negative, and mixed class. After re-

[1] http://www.lemurproject.org

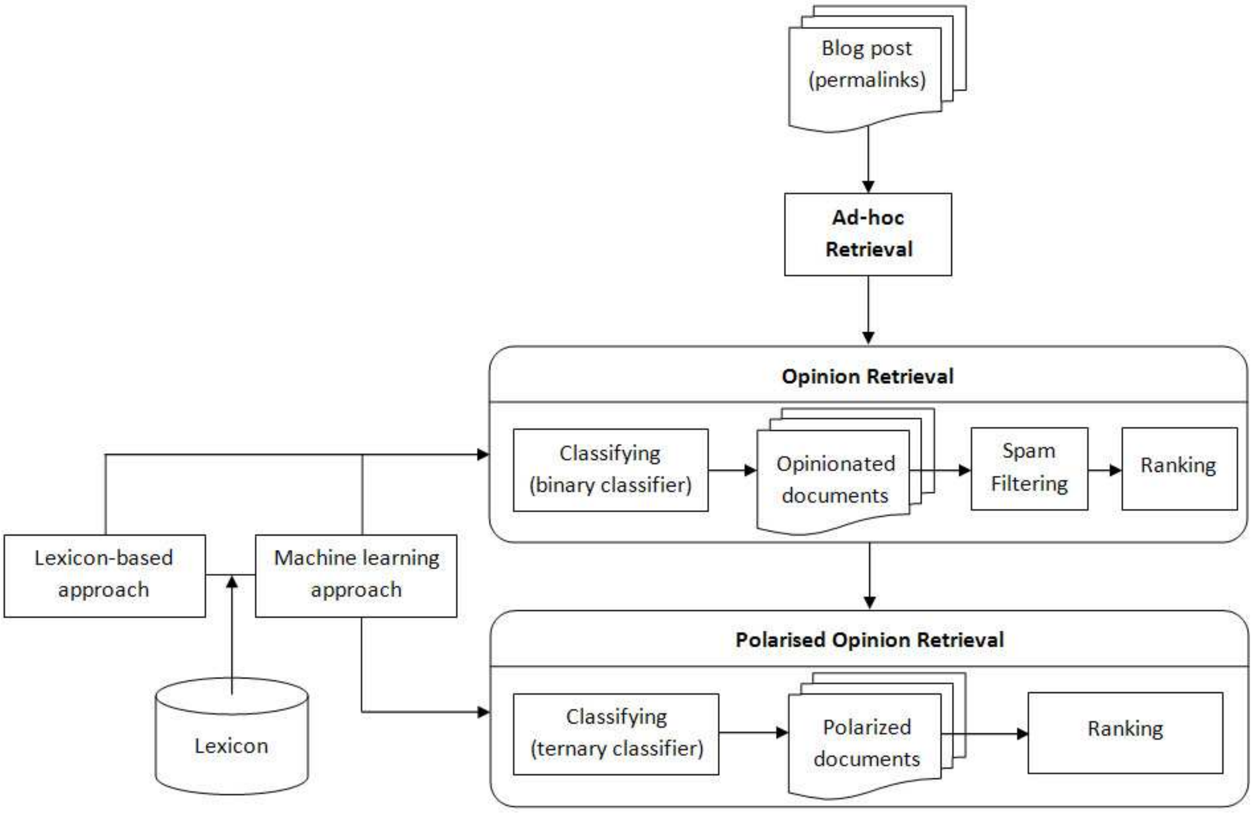

Figure 1: System structure of Opinion Finding task

moving mixed documents, we retained positive and negative documents for ranking. Ranking score is the linear combination of the polar score predicted by the ternary classifier and the opinion score estimated from the Opinion Retrieval subtask.

2 Ad-hoc Retrieval

2.1 Indexing

The Blog06 collection is used again in TREC 2008. For all the three mentioned subtasks, we indexed only permalinks as the retrieval component. At the pre-processing step, we discarded all HTML tags and redundant scripts from permalink documents. Porter stemming and standard stop-word removing are early applied for indexing but they slightly harmed the preliminary retrieval results. Therefore, we did not use neither stemming nor stop-word removal at the final indexing version.

2.2 Query Processing

Query expansion is the process of adding terms to the original search query. To study the effectiveness of query expansion for Ad-hoc Retrieval, we empirically exploited two simple techniques as follows:

Cluster-based analysis: The first technique for expanding query comes from the hypothesis that clustering can provide retrieval extra information. We deployed this idea by using Clusty search engine[2]. Clusty is a text clustering search engine allows grouping search results into folder topics. Empirically, we issue an original query to Clusty and take the titles of top *n* clusters[3] as expanded terms.

Pseudo-relevance feedback: The intuition of pseudo-relevance feedback is that top ranked retrieved documents would be the most relevant ones to the given query. Consequently, those documents might contain terms which are highly related to the topic. For simply examining this method, we se-

[2] http://clusty.com

[3] In Clusty, clusters are sorted by their number of documents

Cluster-based analysis	Pseudo-relevance feedback	Using topic's description
`Windows Vista` Microsoft Software Operating system (*top 3 clusters's title are added to the original query*)	`Windows Vista` Microsoft Software beta (*top 3 most frequent terms of top 50 ranked retrieved documents are added to the original query*)	`Windows Vista` Find opinion Microsoft operating system any features

Table 1: Example of processed queries for the original query `Windows Vista`.

lected *n* most frequent terms in top ranked documents and add them to the original query.

Using topic description has been reported to be marginally beneficial to opinion finding task (Ounis et al., 2006). For this analysis, we used terms in the topic description after removing standard stop words as expanded terms for the original query. Given query `Windows Vista`, Table 1 shows an example of processed queries.

2.3 Language Model

As mentioned earlier, we empirically adopted several out-of-the-box models for Ad-hoc Retrieval. According to our experiments, Kullback-Leibler (KL) divergence was shown to be the best-performed model comparing to other ones. KL is a statistical language model which scores and ranks documents by the KL-divergence (i.e, relative entropy) between the query language model and the document language model (Lafferty and Zhai, 2001). Since ranking documents is of interests, KL-divergence is rewritten as the cross entropy of the query model with respect to the document model. For the reason of performance, we used KL model for both of two submitted baseline runs. Additionally, we applied Bayesian smoothing method using Dirichlet priors with default prior parameter set to 1000.

3 Opinion Retrieval

So far, two effective approaches to detect opinionated documents are lexicon-based approach and machine learning approach. In this work, we employed both of these two approaches for identifying opinionated blog posts and ranking them. The intuition here is that individual approach likely retrieves the different set of documents; the combined system hence performed better due to the increase of recall.

3.1 Lexicon-based Approach

In general, lexicon-based approach starts at constructing a dictionary of terms indicating opinion. The opinionated documents are then decided if opinion terms occur in those documents. Following the general framework, we first compiled a list of opinion words from several sources:

General Inquirer: General Inquirer (Stone et al., 1966) is a manually-constructed lexicon that consists of many semantic and emotional categories. We selected the words within opinion-related categories[4] and added them to the final opinion-word list.

Word Net: Starting with a small set of annotated words (seeds), we iteratively looked for synonyms and antonyms of seeds in the WordNet (Miller, 1992). At each iteration, the newly found words were added to the seed list for the next searching iteration. This process was stopped when there was no new word to be found. Final expanded word set was concatenated to the opinion-word list.

Wilson Word Set: This word set is constructed by Wilson (Wilson et al., 2003), which includes subjective clues for identifying opinionated sentences. We concatenate this word set to the final opinion-word list.

After removing duplicate words from the final opinion-word list, we obtained a dictionary of 29,876 words. The weights of words were trained on the assessment for 100 topics in 2006 and 2007. Empirically, the assessed blog posts were split into *opinionated (O)* and *non-opinionated (N)* set. Each word in the dictionary was then assigned a weight given by:

[4] We considered the categories `Positive, Negative, Arousal, Emotion, Feel, Pain, Pleasure, Virtue, Self, Our` as opinion-related ones

$$weight(w_i) = \frac{P(w_i|O) - P(w_i|N)}{P(w_i|O) + P(w_i|N)} \quad (1)$$

Equation 1 is inherited from (Dave et al., 2003) where $P(w_i|O)$ (resp., $P(w_i|N)$) is estimated by the occurrences of w_i in O (resp., N) over the occurrences of all tokens in O (resp., N).

Using the weighted dictionary, a document retrieved from ad-hoc retrieval is scored as follows:

$$score_{LE}(d) = \sum_{w_i \in d} \frac{tfidf(w_i).weight(w_i)}{nearest(w_i,topic)} \quad (2)$$

where $nearest(w_i,topic)$ is the distance (by word) between opinion word w_i and the original topic. Document d is determined to be opinionated if $score_{LE}(d)$ is positive and non-opinionated otherwise.

3.2 Machine Learning Approach

In the context of machine learning, detecting opinionated documents can be considered as a binary classification task. We thus train a binary classifier on the last-two-year assessment. SVM^{light} package[5] is employed for the classification task. Further believing that machine learning can benefit from the opinion-indicated terms, we used opinion words in the compiled dictionary for LE instead of all lexical unigrams as training features.

3.3 A Hybrid Method

Opinion Retrieval subtask requires ranking opinionated documents after identifying their subjectivity. Considering that the opinion retrieval subtask contains classifying step and ranking step, we adopt a hybrid method for each step. At classifying step, the documents classified as non-opinionated by both of LE and ML are removed. At ranking step, both LE and ML scores are combined with topic relevance score to produce final ranking score as below:

$$score_{OR} = \lambda_1 score_{TR} + \lambda_2 score_{LE} + \lambda_3 score_{ML} \quad (3)$$

where $score_{TR}$, $score_{LE}$, $score_{ML}$ are the topic relevance score of ad-hoc retrieval, the opinion score estimated by equation 2 and the output score of binary SVM classifier respectively. In our experiments, the weights for each component score are heuristically tuned to maximize the MAP of opinion ranking.

3.4 Spam Filtering

Since splogs cause negative effects to retrieval, we adopt spam filtering method similar to the one in (Mishne, 2006). We trained naive Bayesian classifier to detect splogs. The training spam data was collected by querying casino to Google web search engine[6].

4 Polarized Opinion Retrieval

Polarized Opinion Retrieval can be referred to polarity classification subtask in Blog Track 2007 (Macdonald et al., 2007). Ranking polar opinionated documents (*positive* and *negative* ones) is a new issue this year. As mentioned earlier, although mixed documents are not required to be retrieved, identifying them is deemed to make the retrieval accurate. Due to this intuition, we employed machine learning approach for dealing with this task since lexicon-based approach turned out to be poor-performed at preliminary experiments.

4.1 Classifying Polar Opinionated Documents

In our experiments, we used SVM^{multiclass} in SVM package for training a ternary classifier. The training corpus was the last-two-year assessments for 100 topics. The features for classification were still opinion words in the compiled dictionary mentioned in section 3.1. At the classifying step, trained classifier categorizes opinionated documents into *positive*, *negative*, and *mixed* category.

4.2 Ranking Polar Opinionated Documents

At ranking step, *mixed* opinionated documents are ruled out, only *positive* and *negative* opinionated ones are retained for ranking. The final ranking score is the linear combination of opinion ranking score and polar score of ternary SVM classifier.

$$score_{POR} = \lambda_4 score_{OR} + (1 - \lambda_4).score_{SVM} \quad (4)$$

[5] http://svmlight.joachims.org/

[6] http://www.google.com

Run	**Description**	**MAP**	**R-prec**	**P@10**
kunlpKLtt	title only	**0.2713**	**0.3544**	0.5800
kunlpKLtd	title+description	0.2666	0.3465	**0.6567**

Table 2: Results of Ad-hoc Retrieval.

Expansion techniques	**MAP**	**R-prec**	**P@10**
None (title only)	**0.2713**	**0.3544**	**0.5800**
Cluster-based analysis	0.2089	0.2919	0.5240
Local-relevance feedback	0.2105	0.2841	0.4720

Table 3: Effects of query expansions for Ad-hoc Retrieval.

Run	**Description**	**MAP**	**R-prec**	**P@10**
kunlpKLtt	Baseline	0.1991	0.2799	0.3820
kunlpKLttOs	Hybrid model ($\lambda_1 = 0.3, \lambda_2 = 0, \lambda_3 = 0.7$)	0.2234	0.3045	0.5553
kunlpKLttOc	Hybrid model ($\lambda_1 = 0.3, \lambda_2 = 0.1, \lambda_3 = 0.6$)	**0.2285**	**0.3138**	**0.5600**
kunlpKLtd	Baseline	0.1953	0.2739	0. 4553
kunlpKLtdOs	Hybrid model ($\lambda_1 = 0.3, \lambda_2 = 0, \lambda_3 = 0.7$)	0.2186	0.3030	0.5500
kunlpKLtdOc	Hybrid model ($\lambda_1 = 0.3, \lambda_2 = 0.1, \lambda_3 = 0.6$)	**0.2191**	**0.3037**	**0.5620**

Table 4: Results of Opinion Retrieval.

Spam filtering	**Average MAP**	**Average R-prec**	**Average P@10**
No	0.2193	0.3013	0.5400
Yes	**0.2224**	**0.3069**	**0.5563**

Table 5: Effects of spam filtering for Opinion Retrieval.

Run	**Description**	**MAP**	**R-prec**	**P@10**
kunlpKLttPc	Positive ranking ($\lambda_4 = 0.8$)	**0.1454**	**0.2153**	**0.3329**
	Negative ranking ($\lambda_4 = 0.8$)	0.1229	**0.1853**	**0.2754**
kunlpKLtdPc	Positive ranking ($\lambda_4 = 0.8$)	0.1361	0.2086	0.3262
	Negative ranking ($\lambda_4 = 0.8$)	**0.1234**	0.1846	0.2718

Table 6: Results of Polarized Opinion Retrieval.

5 Results and Submissions

We submitted two baseline runs for Ad-hoc Retrieval subtask. Both of them were based on KL-divergence model. Whereas `kunlpKLtt` is the title-only run, `kunlpKLtd` is the title-description run. Table 2 shows the evaluation of these two baseline runs for 150 topics from Blog Track 2006 to 2008. Interestingly note that using description is effective to improve the early precision of ad-hoc retrieval (P@10 is boosted at 13.2% from 0.58 to 0.6567). Additionally, table 3 demonstrates the effects of query expansion techniques for ad-hoc retrieval. It can be shown that neither cluster-based analysis nor pseudo-relevance feedback helped increasing performance.

From each baseline run, we generated two runs based on the hybrid model for Opinion Retrieval subtask (in two runs, parameter λ is empirically adapted to optimize MAP of opinion ranking). Table 4 shows the evaluation of four submissions after filtering splogs. The experimental results showed that the proposed hybrid method remarkably improved opinion retrieval performance over baseline (the best run boosted MAP at 14.8% from 0.1991 to 0.2285). In addition, spam filtering conducted a marginal contribution for opinion retrieval (our experiment resulted in an average increase of MAP at 1%, as shown in table 5).

For the Polarized Opinion Retrieval, we submitted two runs corresponding to the two best runs of Opinion Retrieval (i.e., `kunlpKLttOc` and `kunlpKLtdOc`). Table 6 shows the evaluation of these two runs with respect to positive ranking and negative ranking. It can be shown in detail from table 6 that Polarized Opinion Retrieval is strongly dominated by the underlying Opinion Retrieval results.

6 Conclusion

This paper described our approaches for Opinion Finding task at TREC Blog Track 2008. For Ad-hoc Retrieval, KL-divergence model achieved the best results among the others. Utilizing description of topic was shown to be helpful for boosting early precision of Ad-hoc Retrieval. For Opinion Retrieval, hybrid model of lexicon-based approach and machine learning approach was proposed to detect and rank opinionated documents. Spam filtering slightly helped improving the performance of Opinion Retrieval. For Polarized Opinion Retrieval, machine learning approach was employed for predicting polar opinionated documents. Polar score estimated by classifier was then linearly combined with opinion score to produce the final ranking score of those polar opinionated documents.

References

C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC-2007 Blog Track. *Proceedings of the 16th TREC 2007.*

G. A. Miller. WordNet: a Lexical Database for English. *Proceedings of the workshop on Speech and Natural Language, ACL 1992.*

G. Mishne. Multiple Ranking Strategies for Opinion Retrieval in Blogs. *Proceedings of the 15th TREC 2007.*

I. Ounis, M. D. Rijke, C. Macdonal, G. Mishne, and I. Soboroff. Overview of the TREC-2006 Blog Track. *Proceedings of the 15th TREC 2006.*

J. Lafferty, and C.Zhai. Document Language Models, Query Models, and Risk Minimization for Information Retrieval. *24th ACM SIGIR 2001.*

K. Dave, S. Lawrence, and D. M. Pennock. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. *WWW 2003.*

P. J. Stone, and D. C. Dunphy A Computer Approach to Content Analysis *MIT Press 1966.*

T. Wilson, D. R. Pierce, and J. Wiebe. Identifying Opinionated Sentences *ACL 2003 on Human Language Technology*