NTU at TREC 2007 Blog Track

Kevin Hsin-Yih Lin and Hsin-Hsi Chen

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan, 106

E-mail: hylin@nlg.csie.ntu.edu.tw; hhchen@csie.ntu.edu.tw

Abstract

We participated in the Opinion Retrieval Task and the Polarity Subtask. An SVM classifier was used to determine the opinion polarities of documents. We found that the opinion mean average precisions for the runs using the SVM classifier is better than the opinion mean average precisions for the runs produced solely by the TFIDF retrieval model.

1. Introduction

The Blog track offers several tasks this year. We participated in the Opinion Retrieval Task and the Polarity Subtask. The Opinion Retrieval Task requires us to retrieve documents which are both relevant to a topic and contain opinions. The Polarity Subtask requires us to label the opinion polarities of documents. The possible polarities are negatively opinionated, positively opinionated and mixed polarity.

Our approach to solving these two problems is to use the TFIDF retrieval model to retrieve topic-relevant documents and then apply a support vector machine (SVM) classifier on these documents to identify their polarities. We used the data from TREC 2006 Blog Track's Opinion Retrieval Task as the training data. Our results show that the opinion mean average precisions (MAP) for runs not classified by the SVM classifier are slightly lower than the MAP for their corresponding runs classified by SVM classifiers.

2. Method

Our method for the Opinion Retrieval Task consists of two steps: retrieving topic-relevant documents and ranking the opinionated retrieved documents higher. We used the Lemur toolkit as the search engine for retrieving topic-relevant documents.[1] The information retrieval model was TFIDF. At most one thousand documents were retrieved for each

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

topic.

After obtaining the topic-relevant documents, we classified them into four categories: no opinion, positively opinionated, negatively opinionated, and mixed polarity. Documents belonging to the latter three categories were relocated to the front of the document list.

The LibSVM toolkit was used as our SVM classifier.[2] An annotated corpus was required to train the classifier, so we used the Blog06 collection combined with TREC 2006's Blog Track relevance judgment file as the training corpus. To construct the training corpus, we first retrieved the documents that have the opinion tags of 1, 2, 3 or 4. Each of these documents is a training instance. Features were extracted from the documents' textual contents. We did not use entire documents. Instead, we used only certain sentences from the documents. For each document, we retained only the sentences that contain at least a word from their corresponding topic. The sentences immediately following these sentences were also kept. Features were extracted from this collection of sentences.

The features used were punctuations, words and emotion words. The emotion words are a set of words provided by the Internet General Inquirer's website.[3] We used binary feature weights. That is, if a feature appears in a training instance, then the feature has the weight of 1. The feature has weight 0 otherwise.

For the learning phase of the SVM classifier, we performed 5-fold cross-validation on the training corpus to estimate the best cost parameter for the linear kernel.

After the SVM classifier is generated, we apply it on 2007's topic-relevant documents retrieved by the Lemur toolkit. As with the training corpus, we did not extract features from all the contents of the topic-relevant documents. Only sentences containing words from their corresponding topics and the sentences immediately following the aforementioned sentences were retained for feature extraction. If the SVM classifier labels a topic-relevant document as having an opinion (i.e., negatively opinionated, positively opinionated, or mixed polarity), the document is moved to the front of the list. Hence, the initial document list returned by Lemur was conceptually divided into two sub-lists: the list containing documents with opinions followed by the list containing documents without opinions. Within each list, the documents were ranked by their TFIDF relevance score.

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[3] http://www.webuse.umd.edu:9090/

Table 1: Opinion Retrieval Task Results

Run ID	Query Type	Opinion-Finding	Topicrel MAP	Opinion MAP
NTUManual	Manual	No	0.3051	0.2384
NTUManualOp	Manual	Yes	0.2940	0.2393
NTUAuto	Automatic	No	0.2901	0.2254
NTUAutoOp	Automatic	Yes	0.2870	0.2282

Table 2: Polarity Subtask Results

Run ID	R-Accuracy
NTUManualOpP	0.1161
NTUAutoOpP	0.0967

3. Results

Table 1 shows our results for the Opinion Retrieval Task. The Query Type column indicates whether queries were manually-generated or automatically-generated. We create manual queries by removing general words from the topics. For automatic queries, all the words in the topics are used. The Opinion-Finding column indicates whether document opinion classification was used. In other words, if this value is “no” for a run, then the run is generated solely by the TFIDF retrieval model.

In Table 1, we see that manual queries have higher Topicrel MAPs and Opinion MAPs than automatic queries. When the Opinion-Finding function is turned on, Topicrel MAPs drop, but Opinion MAPs increase.

Table 2 shows our results for the Polarity Subtask. The NTUManualOpP run is generated by labeling the documents listed in NTUManualOp as negatively opinionated, positively opinionated, or mixed polarity. Similarly, the NTUAntoOpP run is generated by labeling the documents listed in NTUAutoOp with opinion categories. The results produced by using manual queries have better R-Accuracy than the automatic queries.

4. Conclusion

In this study, we used the SVM classifier to determine the opinion polarities of documents. Only very simple textual features were used. It may be possible to improve the performance by using more complicated features.