Knowledge Transfer and Opinion Detection in the TREC2006 Blog Track

Hui Yang
Language Technologies Inst.
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, 15213
huiyang@cs.cmu.edu

Luo Si[1]
Dept. of Computer Science
Purdue University
West Lafayette, IN 47907
lsi@cs.purdue.edu

Jamie Callan
Language Technologies Inst.
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, 15213
callan@cs.cmu.edu

Abstract
The paper describes the opinion detection system developed in Carnegie Mellon University for TREC 2006 Blog track. The system performed a two-stage process: passage retrieval and opinion detection. Due to lack of training data for the TREC Blog corpus, online opinion reviews provided in other domains, such as movie review and product review, were used as the training data. Knowledge transfer was performed to make the cross-domain learning possible. Logistic regression ranked the sentence-level opinions vs. objective statements. The evaluation shows that the algorithm is effective in the task.

Introduction
The Blog track is a new task in the TREC 2006 evaluation. The main task of the track is "opinion detection" in the domain of the online blogs posted during the period of Dec 2005 to Feb 2006. The posts and the comments are from Technorati, Bloglines, Blogpulse and other web hosts. The system developed in Carnegie Mellon University for the opinion detection task consists of the modules described below.

Data Preprocessing
The data from NIST are mainly xml files with tags similar to previous TREC web collections, such as WT10G or GOV2. Three types of files are provided by NIST: permalinks (html documents containing the posts), RSS feeds, and blog homepages (html documents containing the homepages of the feeds). Permalinks contains the actual content of the corpus, and are the main target of this task. Indexing, retrieval and opinion detection are all performed based on permalink documents. Further study should involve RSS feeds since they reveal the structure and network of multiple blog posts.

Due to messy nature of online html, data cleaning is an important preprocessing step. Two approaches were tried. The first utilized the built-in html file cleaning functions of the latest Indri 2.3.1 toolkit [1]. Additional preprocessing was done to handle stylesheets and javascript, which were not handled by the current version of Indri. The index was then built based on the "trecweb" format supported by Indri. The unit for indexing and retrieval is one permalink document, i.e., one blog post with its following comments. However, it was soon realized that taking the raw html files and throwing them into Indri to index limits the flexibility of gathering more information from the raw text, for example, sentence structure, paragraph information, part-of-speech tagging, etc, which could be important for opinion detection in later stages. These could all be done by creating more functions in Indri, however, due to the amount of programming effort and time constraints, the first approach was discarded.

The second approach was to transform the html files as closely as possible into regular text files. This was done by several steps.

Removing HTML tags, scripts, stylesheets: A wrapper was created on top of a tool called "striptags" from the REAP project [2]. The text documents looked much neater; however, there were still advertisements, text from side bars and menus floating around. These are all noise in the main text. To remove them another module with machine learned patterns could possibly remove them. However, such noise would also be filtered out automatically by the retrieval and opinion detection in the later stages, hence leaving them in caused little harm to the final results.

[1] This work was done while the author was at the Language Technologies Institute at Carnegie Mellon University.

Removing Non-english characters: The TREC 2006 Blog corpus contains non-English posts. Characters with ASCII code less than 32 and greater than 126 were removed from the corpus.

Sentence Splitting: A modified version of UIUC's sentence splitter [3] was used to annotate the corpus with <s> and </s> tags that identified the beginning and end of each sentence.

Creating Artificial Paragraphs: Original line breaks from the text were reserved as segmentations of paragraphs. Moreover, for long original paragraphs, an around-100-word paragraph break was introduced with no crossing of the sentence boundaries.

Removing Dummy Sentences: If a sentence contained only punctuation, or just a single number, it was removed. This step filtered out form counters largely available on the Web documents. Moreover, if within a sentence, any word had an occurrence of more than N times (N=20 in these tests), that sentence was removed. This step filtered out advertisement and web category anchor text, which were not important content of the Web blog and hence are irrelevant to any potential query

<DOC> <TEXT>
<PARAGRAPH>
<s>blah blah blah</s>
<s>blah blah blah</s>
…
</PARAGRAPH>
<PARAGRAPH>
…
</TEXT> </DOC>
.

Query Formulation

The topic file that NIST provided contains topics each with the title, description and narrative parts. Based on the title field, the topics could be classified into 6 categories [Figure 1]:

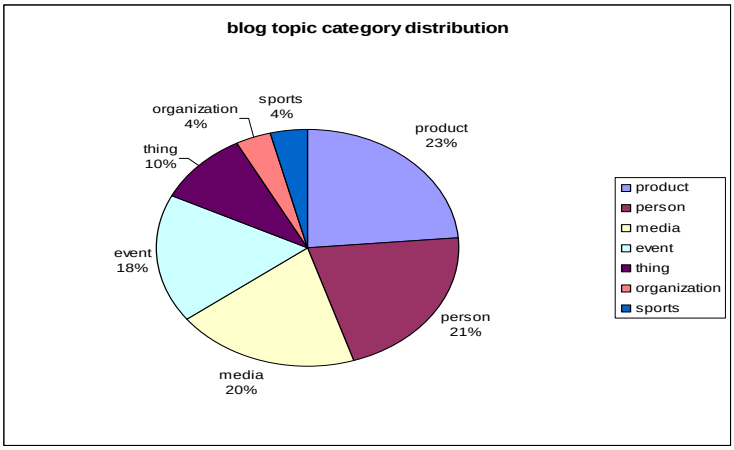


Figure 1: Topic Category

The TREC Blog track required at least one run using just the title as the query. The Minipar parser [1] was applied to identify part-of-speech for the title, description and narrative terms. To create the query, only nouns and adjectives were extracted as query terms. A dictionary () of positive and negative verbs and adjectives was downloaded from the web, and manual selection was used so that the list was not long enough to increase retrieval time too much.

Based on the nouns and adjectives from the title, description and narrative as well as the sentiment word dictionary, four types of queries were formulated by the system:

- The title query;
- The title query and the nouns and adjectives from the descriptions;
- The title query and sentiment words from the dictionary; and
- The title query, the nouns and adjectives from the descriptions and sentiment words from the dictionary.

Table 1: Opinion Word Dictionary

#positive verb	#negative verb	#positive adjective	#negative adjective
Love, like	Hate, dislike	Good, best, better, happy, extraordinary, successful, glad, desirable, worthy, remarkable, funny, lovely, entertaining, decent, beautiful, fascinating, brilliant, gorgeous, perfect, nice, fantastic, impressive, fabulous, amazing, desirable, excellent, great, awesome, splendid, distinctive	Bad, awful, suck, worse, worst, poor, annoying, stupid

The Indri queries automatically generated by the system were structured queries (see Indri query language in [1]) which treat "title" as a phrase (an ordered window) , "description" words as expanded terms, and "dictionary" words as terms within an unordered window to "title" words. Here are examples for the above four categories:

- #combine[paragraph](#3(march of the penguins))
- #combine[paragraph](#3(march of the penguins) documentary film)
- #combine[paragraph](
  #uw15(#3(march of the penguins) love)
  #uw15(#3(march of the penguins) like)
  …
  #uw15(#3(march of the penguins) great)
  #uw15(#3(march of the penguins) awesome))
- #combine[paragraph](
  #uw15(#3(march of the penguins) love)
  #uw15(#3(march of the penguins) like)
  …
  #uw15(#3(march of the penguins) great)
  #uw15(#3(march of the penguins) awesome)
  documentary film)

Indexing and Passage Retrieval

Two kinds of Indri indexes were built, one with stopword removal and krovetz stemming, and the other with neither of them. The indexes were built over the paragraphs created in the earlier stage.

Passage retrieval was preferred to document retrieval in our system. For opinion retrieval, it was believed that only a small amount of text, probably a paragraph in a post, was needed to show the loves and hates. Moreover, smaller units help to focus on the opinions instead of objective sentences. Therefore, the accuracy of opinion detection should be higher by using passage retrieval than using document retrieval. Another concern was that there could be multiple opinions about one topic in a blog post. If document is used as the unit, there is no chance to identify different opinions within the same document. However, in the TREC blog evaluation, document *is* used as the unit to evaluate, i.e., as long as a document contains an opinion, it should be returned in the result. We believe that a finer unit than document is necessary for a more accurate evaluation of opinion detection.

Since there are many duplicated documents in the Web blog collection, duplicate pruning was conducted after documents were retrieved. First, 1000 paragraphs were retrieved for each query, then the duplicate detection module removed any exact duplicates within the retrieved results. The results list was extended, as necessary, so that 1000 passages remained after duplicate removal. Duplicate removal helped the system focus on unique passages and hence improve the quality of retrieved results.

Knowledge Transfer and Opinion Detection

The opinion detection task was considered as a binary text classification problem. Logistic regression was used to rank the sentences from the retrieved passages. The feature space was represented as $F = \{w_1, w_2 ..., w_n\}$, where w_i 's n-gram features, n is the feature space size. A sentence *S* was

represented as $S = \{X_1, X_2 ..., X_n\} \in \{0,1\}^n$, where X_i is the binary value indicating presence of feature w_i in the sentence. The task is to predict labels *Y* for test set sentence *S*, *Y*=1 when *S* is an opinion and *Y*= -1 when *S* is an objective sentence. Logistic regression model is trained based on *m* labeled training set examples $D = \{(s_1, y_1), (s_2, y_2) ..., (s_m, y_m)\}$. The likelihood function is:

$$L = \Pr(Y = 1 \mid S = s, \theta) = \frac{1}{1 + \exp(-\theta^T s)} \quad (1)$$

where θ is the parameter vector. Gaussian prior $N(0, \sigma^2 I)$ are assumed for the parameters, in particular, for components in the parameter vector, independence is assumed among them. For every component, zero-means and equal variances are applied. The estimated parameter vector θ is obtained by maximizing log-likelihood function via maximum a posterior (MAP):

$$\theta_{\max} = \arg\max_{\theta} (\sum \log \Pr(Y = y \mid S = s, \theta) - \frac{1}{2\sigma^2} \left\| \theta \right\|_2^2) \quad (2)$$

Due to the lack of training data, two training data sets in different domains were used. The first dataset was movie review data provided by Pang and Lee (http://www.cs.cornell.edu/People/pabo/movie-review-data/). It includes 5000 subjective sentences and 5000 objective sentences. The subjective sentences are sentences expressing opinions about a movie. The objective sentences are descriptions or the storytelling of a movie. The second source was the customer review data provided by Hu and Liu (http://www.cs.uic.edu/~liub/FBS/FBS.html). It contains 4258 sentences in total with 2041 positive examples and 2217 negative examples. The customer reviews are from Amazon.com about 5 electronic products including digital cameras, DVD players and jukeboxes.

The training data is from movie and product review; however the testing set is from the blog posts. The domain of training set data is different from that of the testing data. The technique of transfer learning [6][7] is necessary in this case. The knowledge learned from one domain should be able to help the leaning in another domain, as long as they are similar task. Hence opinion detection with knowledge transfer is preferred in this case.

A simple approach of knowledge transfer was taken. Based on the two training datasets, movie review and product review, features highly ranked in both domains are considered as common features for the task of opinion detection across all opinion-related domains. In another word, they are considered as domain-independent features. The rest of the features in the training data were then removed in the feature selection step. For example, in the movie review unigram model, “film” has a high occurrence of 1006, and “movie” has a high frequency of 733. Both of them are within the top 35 features, however they are not domain-independent, which means that they are not useful for classifying the opinions and non-opinions in other domains. Those domain-dependent high frequency features were removed from the vocabulary.

Another source was sentences extracted and annotated from the 2006 TREC Blog corpus itself by creating training topics manually. There were 1201 positive examples and 1240 negative examples manually drawn from the training topics. The training topics used to create the training data were: “Steelers”, “Christmas”, “Mr. and Mrs. Smith”, “Harry Potter”, “Condoleezza Rice”, “Canon camera”, “China rise”, “Harvard University”, “Hash browns”, “Seattle”, and “Zoloft”.

Bayesian Logistic Regression toolkit (BBR [5]) was used to rank each retrieved sentence. The passage score was generated by averaging scores of sentences within a passage. Unigram, bigram and trigram models were built for regression. Another feature used in the experiment was the percentage of adjectives in a sentence by parsing the sentence for its part-of-speech tags. It is important because many opinions describe things using adjectives. For example, “the camera is great,” “It is an awesome movie,” And “Here's the brief synopsis: the phone is tiny, cute, feels kind of plastic-like.” The main features used in the runs submitted to TREC summarized in Table 2.

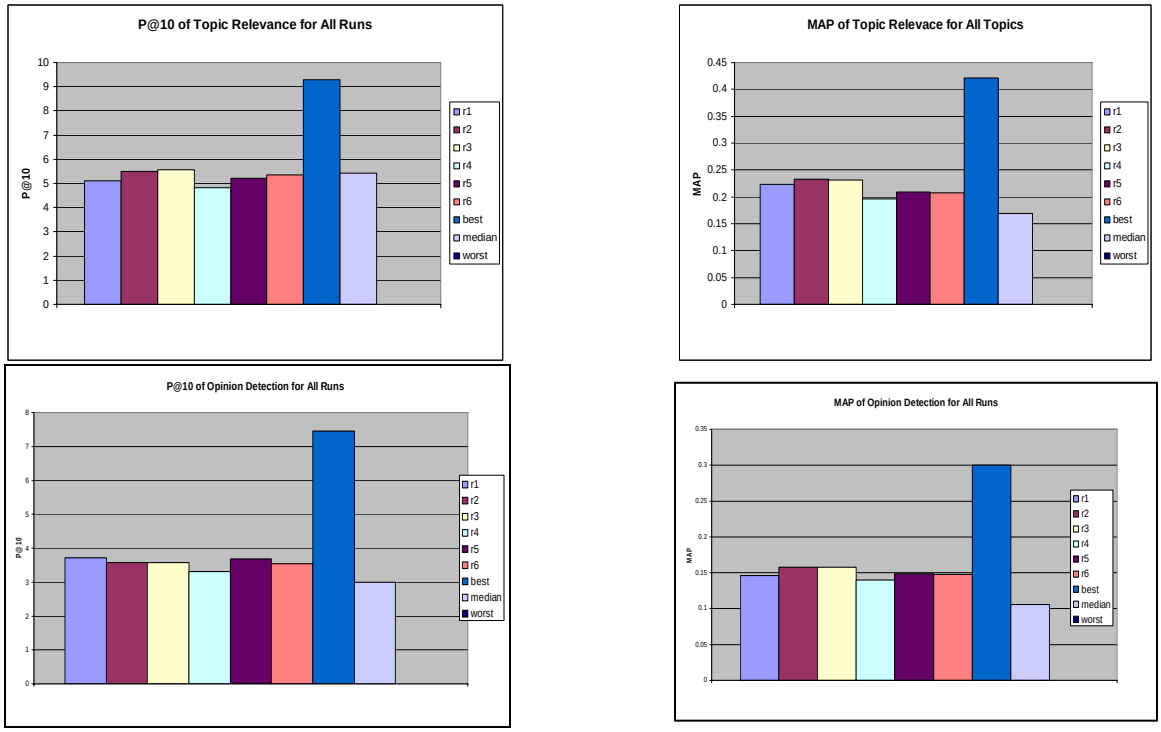
Table 2: Description of Submitted Runs

Run tag	Submit Priority	Description
blog06r1	6	Title query, unigram model + adjective percentage
blog06r2	3	Title query, bigram model + adjective percentage
blog06r3	5	Title query, trigram model + adjective percentage
blog06r4	4	Title + description + opinion words, unigram model + adjective percentage
blog06r5	1	Title + description + opinion words, bigram model + adjective percentage
blog06r6	2	Title + description + opinion words, trigram model + adjective percentage

Evaluation and Experimental Results

TREC 2006 Blog task contains 50 topics. For each topic, evaluation was done by calculating the mean average precision (MAP), R-precision and precision at 10 (P@10) of returned passages. Early precision is important. Figure 2 shows the MAP and P@10 of opinion detection task and the topic relevance for the first stage of passage retrieval. The top right and top left panel of Figure 2 show average MAP and P@10 of topic relevance respectively for all 50 topics. The 6 runs are compared with the "best" artificial run (created by using the MAP or P@10 of the best performing system for each topic), and the median artificial run (created by using the MAP or P@10 of the median performing system for each topic).

Figure 2: MAP and Precision @10 for topic relevance and opinion detection



Based on the results, the following observations can be made.

- Our best run was blog06r2. For both topic relevance and opinion detection, it gives the highest MAP. This run is bigram model with no query expansion. However it was not the run selected for submission to TREC.
- The MAPs of the 6 runs are above the median run by 20%-53.3% for topic relevance, 40%-60% for opinion detection.
- The P@10 of some runs are below the median run for topic relevance, however, the P@10 for all the runs are beyond the median run for opinion detection. This indicates that our knowledge transfer and logistic regression model is effective to recognize opinions from retrieved passages.

- The fact that Bigram and trigram models succeed unigram model is not surprising, since bigram and trigram models capture phrases to express opinions, for example, "I love", "is awesome", "really like it", which is more reliable features than just a single word.
- Out of our expectation, query formulation plays a useless role in passage retrieval. The topic relevance for run blog06r1, blog06r2 and blog06r3 are better than the rest three runs, which make use of query expansion with topic description and opinion words. This shows that the description words in the topic given by NIST are not effective in finding the relevant passages in Blog corpus. There may be some vocabulary mismatch in the description and corpus. Since the query expansion is done with both description words and opinion words, the improvement introduced by adding opinion words in the queries are covered. The experiments conducted after TREC submission shows that query expansion with opinion words do increase the chances of finding opinions in the passage retrieval stage and hence improves the opinion detection.
- Topic retrieval plays an important part in the final opinion detection results. It constrains the number of passages that are candidate to be opinions. The low recall of the opinion detection is mainly due to low recall of topic retrieval. Useful documents are not retrieved by the search engine from the blog corpus, though we believe our approach of logistic regression with knowledge transfer is effective in this task, the performance of passage retrieval constraints the opinion detection module, which cannot show a more impressive result. It will be the future work to improve the system performance.

Conclusion

This paper showed a two-stage opinion detection system developed for TREC 2006 Blog track. In the frist stage, language model passage retrieval were performed to get the relevant passages to the given topic. In the second stage, knowledge transfer across multiple domains was employed and logistic regression were used as the main algorithm to solve the opinions vs. objective sentences binary text classification problem. Our 6 submitted runs performed not the best, but better than the median of all the submitted runs. The TREC evaluation showed that the knowledge transfer and logistic regression approach are effective for recognizing opinions against objective statements in the blog space.

Acknowledgements

This research was supported by NSF grant IIS-0429102. Any opinions, findings, conclusions, or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsor. The authors are also grateful to Mark Hoy and Jie Lu for helpful discussions for the system development.

References

[1]. Indri search engine package: http://www.lemurproject.org/indri/
[2]. REAP project: http://reap.cs.cmu.edu/
[3]. UIUC Sentence Splitter: http://l2r.cs.uiuc.edu/~cogcomp/atool.php?tkey=SS
[4]. Minipar parser: http://www.cs.ualberta.ca/~lindek/minipar.htm
[5]. BBR: Bayesian Logistic Regression Software: http://www.stat.rutgers.edu/~madigan/BBR/
[6]. Rajat Raina, Andrew Y. Ng, and Daphne Koller, "Transfer Learning by constructing informative priors", Workshop on Transfer Learning in the Nineteenth Annual Conference on Neural Information Processing Systems (NIPS 2005).
[7]. Guillaume Obozinski, Ben Taskar, Michael Jordan, "Multi-task Feature Selection", In the workshop of structural Knowledge Transfer for Machine Learning in the 23rd International Conference on Machine Learning (ICML 2006).