IIT Kharagpur at TREC 2008 Blog Track

Robin Anil, Sudeshna Sarkar
Indian Institute of Technology Kharagpur
robin@cse.iitkgp.ernet.in, sudeshna@cse.iitkgp.ernet.in

25 Oct, 2008

Abstract

This paper describes our opinion retrieval system for TREC 2008 blog track. We focused on five different aspects of the system. The first module is focussed on extracting the blog content out from junk html and thereby decreasing the noise in the indexed content. The second module aims at removing various kind of spam content from real blogs. The third module aimed at retrieving the relevant documents. The fourth module filters out opinionated documents and the fifth one calculated the polarity of the sentiments in the document. The final ranked retrieval runs were based on various combination of settings in each module so as to study the effect of each. For classification of subjectivity and polarity, the predictions we done using a complementary naive bayes classifier

1 Introduction

Web logs (blogs) are a fast growing phenomenon on the World Wide Web (Web) as they allow people to publish their thoughts and opinions on any topic they choose. In July 2006 a blog tracking company, Technorati, Inc., reported that it was tracking 48.8 million blogs worldwide (currently tracking 48.8 million sites and 2.7 billion links 2006), up from 4.2 million in October 2004. There were over 100 Million Blogs in 2007 and growing. Blogs are an informal form of communicating. Blogs created by organizations such as newspapers or those addressing political or popular topics receive thousands of hits per day, however most personal blogs have a far smaller audience. The author of a blog is free use any language required to express their thoughts and opinions. The length of blogs varies from one paragraph to pages of rants, the paragraphs within the blogs also vary considerably. On the other hand, the length of a news article is dictated by the space available. However, one paragraph articles are uncommon. These articles have trained and experienced authors and an editor to ensure high quality writing techniques are used and words are not used out of context or with ambiguous meaning.

Blog posts are often informally written, poorly structured, and filled with spelling and grammatical errors, and feature non-traditional content. Performing linguistic analysis on blogs is plagued by two additional problems: (i) the presence of spam blogs and spam comments and (ii) extraneous non-content including blog-rolls, link-rolls, advertisements and

sidebars. TREC Blog Track 2006 and 2007 saw some good amount of interest in this field. This problem was again posed in the TREC 2008 Blog track with a few modifications.

For the purpose of the TREC evaluation, the participants were working on the Blog06 test collection. The permalinks of documents was used as a retrieval unit. The number of such permalinks was around 3.2 million. Our system of retrieving the documents was made using the Apache Lucene search engine. Lucene was able to index the whole Blog06[6] dataset and retrieved the documents very quickly(˜0.1s). Inorder to decrease the size of the index it was necessary to remove a lot of noise in the HTML. A lot of the documents had malformed html which was corrected using the HTML Tidy utility. We used the qrels of the Blog Track of TREC 2006 and 2007 to train the sentence level subjectivity and polarity classifiers. In the following sections we detail how we structured the whole system.

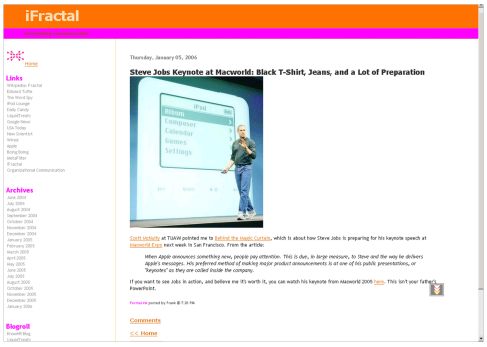
2 Data Preprocessing - Content Extraction

Web pages are cluttered with distracting features around the body of a blog post which distract the user from the content block. These features range from pop-up ads, flashy banner advertisements, unnecessary images, or links scattered around the screen. These objects are quite unimportant and thus add noise to the main content. The noise in the content may create errors while doing document retrieval thus drastically reducing the precision of retrieval. In order to improve the quality of opinion extraction results, we extracted the title and content of the blog post for indexing because the scoring functions and Lucene indexing engine cannot differentiate between text present in the links and sidebars of the blog post. Automatically extracting the actual content poses an interesting challenge for us. Blogs are a bit more structured than random html pages. The parts we are interested are the title, post and comments area of the page. But the random nature and structure of different blog softwares make it quite difficult for simple rule based content extraction. Our approach to html extraction was based on the text to link ratio principle.

The core idea is that a content rich part of the blog will have a higher text/link ratio than the side bars, header, footer and other advertisements. We did a DFS traversal of the HTML tree and calculated the ratio for the whole set of nodes. After this is done the second step starts from the body tag and goes down the tree in breadth first fashion. We do a hill climbing search to find the node with the highest text to link ratio. Since blogs are simpler than other html pages on the internet, we were able to extract the required content in almost all of the cases barring a few($<0.001\%$). In most of the cases the document was retrieved with none or few extraneous content(1-2 sentences long). We also tried using a SVM classifier to predict the content blocks based on the link ratio, tags and html attributes. The result was quite similar to the hill climbing heuristic, but it skipped many important blocks in some of the cases. We needed to index most of the content, so indexing the content with partial noise was preferred to the one where some content blocks are unrecognized. Figure 1 shows the screenshot of a Blog and the extracted content on a sample blog post.

3 Splog Detection

Spam filtering was done based on a seed data set (Kolari, Finin, & Joshi 2006) of 700 positive (splogs) and 700 negative (authentic blog) examples. Each one is the entire HTML content of



Title: iFractal: Steve Jobs Keynote at Macworld: Black T-Shirt, Jeans, and a Lot of Preparation
Post: Steve Jobs Keynote at Macworld: Black T-Shirt, Jeans, and a Lot of Preparation. Scott McNulty at TUAW pointed me to Behind the Magic Curtain, which is about how Steve Jobs is preparing for his keynote speech at Macworld Expo next week in San Francisco. From the article: When Apple announces something new, people pay attention. This is due, in large measure, to Steve and the way he delivers Apple's messages. His preferred method of making major product announcements is at one of his public presentations, or "keynotes" as they are called inside the company. If you want to see Jobs in action, and believe me it's worth it, you can watch his keynote from Macworld 2005 here. This isn't your father's PowerPoint. PermaLink posted by Frank @ 7:20 PM

Figure 1: An example show HTML content extraction from blogs

the blog home-page. The judgment was made based on just the local features or required the in-links and out-links of the pages. Discriminating features included content, out-links, in-links, post time-stamps, blog URL, post comments and blogrolls[1]. We analysed how much the models will be able to minimize misclassification of authentic blogs while maintaining accuracy of the splog classification. Finally we determined that using a bag of words , model provided a decent set of features that did so. All of the models are based on Complement Naive Bayes classifier(Implementation in Apache Mahout[2]) using Bag-of-words, bigrams and trigrams as features. A few shortcomings we faced in this classifier are that the dataset wasn't complete in its coverage of spam. So In order to facilitate better classification, we increased the dataset by manually annotating some splog in the Blog06 dataset itself. Finally we did filtering of offensive content. A major chunk of the Blog06 dataset (˜18%) was identified as spam or offensive and was hence removed.

The Blog06 dataset also contained a lot of non-english blogs. Since our system only dealt with english language opinions it made no sense to keep the non english ones. They were removed by a simple heuristic comprising of the number of occurence of non english characters and the articles(a, an, the etc.) in the document.

4 Document Indexing and Retrieval

Lucene[5], an open-source search engine is a powerful search framework capable of indexing several gigabytes of document data and quickly performing complex searches on that data. Lucene can also process data beyond raw text of the document content. Typically this includes data about the documents that are being indexed, for example, title information, document author, feed url etc. Lucene provides a scoring algorithm that includes this additional data to find best matches to document queries. The default scoring algorithm is fairly complex and considers such factors as the frequency of a particular query term with individual documents and the frequency of the term in the total population of documents.

After pre-processing, blog posts are indexed using Lucene. Lucene internally constructs an inverted index of the documents by representing each document as a vector of terms. Given a query term, Lucene uses standard Term Frequency (TF) and Inverse Document Frequency (IDF) normalization to compute similarity. In addition, the scoring formula can also be tuned to perform document length normalization and term specific boosting. The default parameters were modified while searching the index. In our system we maintained two different indices, one at the sentence level and the other at the document level. We

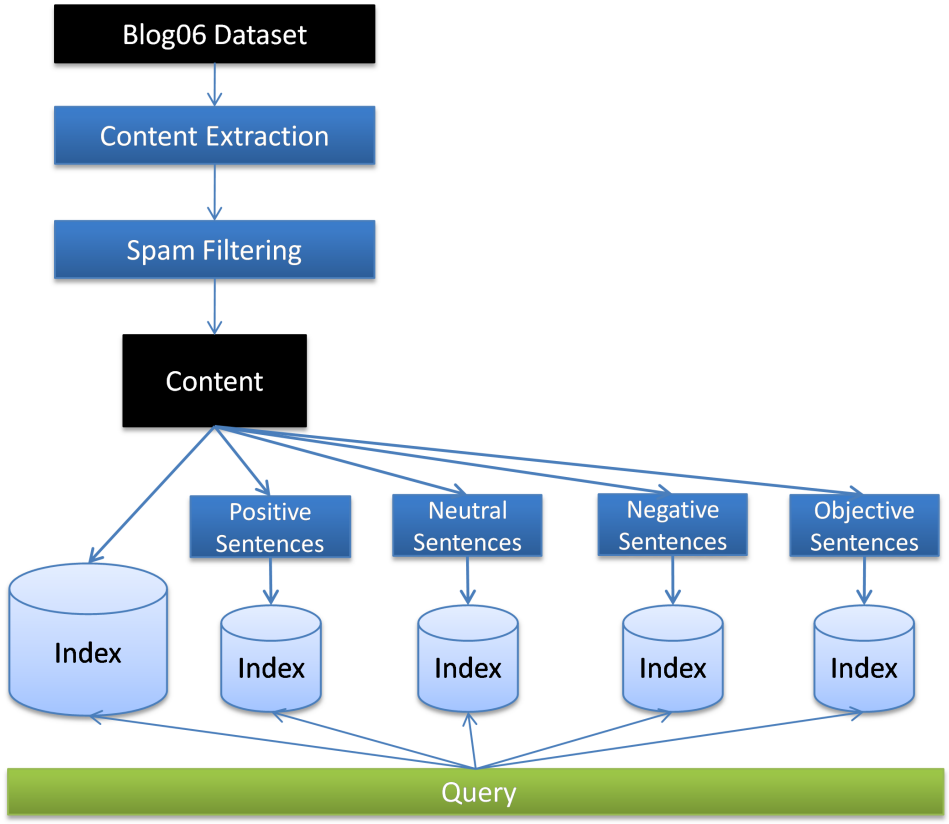


Figure 2: The Overview of the system

ran our opinion detection and polarity detection module through the whole Blog06 dataset and split a document into 4 lists. The lists had only objective, positive, neutral or negative sentences from the given document. We indexed these four group of sentences as a field as well as the text of the whole document. Using the given TREC queries, we queried on each of these 4 fields/indexes as well as the whole document (content field) as shown in figure 2. The final score of the retrieved document was based on the BaseScore(Score from the query on the content field) and the OpinionScore(Score from positive, negative and neutral field). Score from the objective index was given lesser weight. This ensured that the opinionated documents come on top in the ranked retrieval result.

Once the documents are retrieved they are run through a reranking module which does the reranking on the basis of the output for opinion classification and polarity detection modules.

5 Opinion Filtering and Polarity Detection

The Opinion Filtering and Polarity Detection module aims at improving the opinion finding accuracy of the system. This module takes in the ranked result and re-ranks it based on the topic relevance and opinion relevance with respect to the given topic keywords.

The Opinion finding system primarily builds on the works of Pang & Lees EMNLP 2002[4] and Pang & Lees ACL 2004[3]. Pang & Lee (2002)[4] examined whether it suffices to treat sentiment classification simply as a special case of topic-based categorization (with two topics

representing different polarities). They conducted sentimental classification using three standard classification techniques (Nave Bayes, Maximum Entropy, Support Vector Machine). They also examined several factors that make sentimental classification more challenging such as the thwarted expectation problem mentioned in the prior section.

5.1 Data for Subjectivity and Polarity Classification

For the purpose of classification we looked at various datasets available today. Movie reviews are typically subjective and can be classified as either positive or negative opinions. They are thus suitable for our classification evaluation. Our experiments use Pang & Lees movie review datasets[3] from http://www.cs.cornell.edu/People/pabo/moviereview-data/. But the size of the movie dataset(10k sentences) had less coverage over the opinions found in the Blog06 dataset. So we had to augment that system with data from various sources. The final dataset for polarity and opinion finding tasks were extracted from the following sources.

Sentiment Polarity Datasets (v 2.0) The data source was gathered from the Internet Movie Database IMDB archive of rec.arts.movies.reviews. There are 1000 positive and 1000 negative full text movie reviews. The ground truths (actual polarity ratings in this case) were assigned based on Pang & Lees heuristic scripts extracting the first review score from each review such as Thumbs Up/ Thumbs Down and Star Ratings.

Subjectivity Datasets (v 1.0) The data source contains 5000 objective sentences and 5000 subjective sentences. Objective sentences were extracted from Internet Movie Databases plot summaries while the subjective sentences are from Rotten Tomatoes review snippets.

Minekey Opinion Dataset We were able to obtain the database of posted opinions from Minekey (www.minekey.com), an opinion discussion network. The database had over 100K sentences, which are very opinionated. We took these sentences extracted the topic and created a parallel dataset of objective sentences by fetching sentences from wikipedia on the same topic.

Extracting parallel sentences from previous Qrels For each of the opinionated documents in the qrels for the year 2006 and 2007, we extracted all the sentences which has the query keyword(which is relevant to the topic in question) in it. We then created a parallel polarity and subjective-objective dataset by grouping these topic relevant sentences into positive/negative and subjective/objective sets. To minimize the bias towards the topic qrels, we removed the query words from the dataset before running the trainer of the classifier.

6 Results

6.1 Topic Relevance and Opinion Search

In the following paragraphs, we detail how the Mean Average Precision(MAP) of each run compares with respect to the different techniques we have implemented. Even though we will be referencing the opinion finding MAP, the trend in topic relevance MAP follows the former with some degree of correlation.

Run ID	Topic MAP(all)	Topic MAP(50 new)	Opinion MAP(all)	Opinion MAP(50 new)
KGPTITLE1	0.3562	0.3576	0.2671	0.2935
KGPNOSPAM	0.3612	0.3560	0.2720	0.2988

Table 1: The Precision Scores for Baseline runs

The output of the search engine with no postprocessing was the run IITKGPTITLE1. All the other runs are built on top of this base system.

6.1.1 IITKGPNOSPAM

The effect of how spam and offensive content was affecting the precision of the retrieved result was being tested from the baseline. The spam filtering was switched on in IITKGPNOSPAM. This resulted the MAP value to increase in both opinion relevance and topic relevance.

6.1.2 KGPOP

In KGPOP, we tested the effect of the movie review opinion dataset in subjectivity and polarity classification. When a document is fetched, we select the sentences which refer to the topic in question. Then each of these sentences are classified as being subjective or objective. Further we calculated the opinion score based on the number of positive and negative sentences.

6.1.3 KGPPOS1

In the Opinion finding task, the best opinion finding MAP (0.30 in KGPPOS1) on the new topics was achieved from a classifier trained using the opinions from minekey.com as the dataset. This classifier used also POS tags as a feature for classification.

6.1.4 KGPPOS2

Another major problem we test was how well we could find and classify the sentences in the document which actually talked about the topic. In KGPPOS16.1.3 we used only the sentences from the document where the query keywords appeared. In KGPPOS2 we also considered the next sentence and tried classifying its as opinionated or not. We saw a decline in the MAP for that run.

6.1.5 KGPFILTER

A bigger gain in MAP was made on removing offensive/pornographic content which plagued the dataset. Along with the filtering we used the dataset of parallel sentences from the qrels and wikipedia for opinion classification in the run KGPFILTER. The overall MAP for that run was 0.47. But the MAP value of the new topics was hovering around 0.28. This performance hit on the new topics suggested that there can still can be more improvements in subjectivity classification methods using that parallel sentences corpora. In the run KGPBASE4, we used the given baseline-4 and used the same rerank module as in KGPFILTER.

Run ID	Topic MAP(all)	Topic MAP(50 new)	Opinion MAP(all)	Opinion MAP(50 new)
KGPOP	0.3284	0.3103	0.3079	0.2717
KGPPOS1	0.3503	0.3430	0.3148	0.3005
KGPPOS2	0.3558	0.3105	0.3145	0.2784
KGPFILTER	0.4372	0.2172	0.4698	0.2847
KGPBASE4	0.4258	0.3270	0.4799	0.2852

Table 2: The Precision Scores for Opinion Finding runs

Run ID	Pos MAP(all)	Pos MAP(50 new)	Neg MAP(all)	Neg MAP(50 new)
KGPPOL1	0.1270	0.1330	0.0952	0.0990
KGPPOL2	0.2416	0.0982	0.3287	0.0587

Table 3: The Precision Scores for Polarity runs

6.2 Ranked Retrieval of Polarity

6.2.1 KGPPOL1

In this run, we relied on the classification of polarity using movie review dataset. In the classifier we used the bigrams, part of speech tags as features. We also added functionality to identify single and double negations.

6.2.2 KGPPOL2

In this run, we relied on the classification of polarity using parallel sentences extracted from the previous year's qrels. The classifier used the same features as in KGPPOL16.2.1.

6.3 Blog Distillation

6.3.1 FEEDKGP

Our feed distillation extraction was based on the opinion extraction module. We took the top thousand retrieved posts from our system and reranked them on the basis of the count of documents from each feed. This run was based on opinion run using movie review data.

6.3.2 FEEDKGP1

This run was based on the same system as in the run KGPFILTER. The ranking was based on the log of the count of documents from each feed.

Run ID	MAP	R-prec	P@5	P@10	P@20
FEEDKGP	0.1539	0.1330	0.3240	0.2680	0.2570
FEEDKGP1	0.1720	0.2484	0.3600	0.3220	0.2770

Table 4: The Precision Scores for Blog Distillation runs

7 Discussion

Comparing the median scores achieved by other teams in all topics, our score in the new topics is slightly below the overall median(hovering around 0.28). Some of the topics have MAP>0.4(in around 24 topics/50) where as some have a low score. Thus the average came out at 0.28 for these 50 new topics. The primary reason for a low map value is the failure of the retrieval engine to fetch the documents in difficult queries. For eg. when asked for articles "citing wikipedia as the primary source", the lucene was unable fetch relevant documents just based on the keywords. For those topics which returned a good amount of documents, the rerank module was able to pick out relevant as well as the opinionated ones correctly. The MAP value was consistently above 0.4 for them. 20 out of these 50 topics did not retrieve enough relevant articles. These topics haven't been search engine friendly so the lack of documents(in a automated retrieval scenario) from our system became the reason for the low MAP value. The average P@5 was near 0.34 for all topics whose MAP was below 0.25(around 30/150). Many of these topics had less than 30 relevant documents retrieved(out of thousand) while some(around 3) had around 80 documents. These precision scores give us some confidence in the classification accuracy of our opinion filtering module. Our focus for the coming year would be to be able to identify the topic of the query in a much more precise manner. The results blog distillation task depended on the output of the opinion filtering system. That too showed the increase in performance due to the improvement in the opinion filtering module.

In the Polarity task, polarity phrasal features extracted from the previous qrels helped to increase the performance. Using a split index for positive, negative and neutral sentences did proved to be faster way to get decent quality results instead completely relying on post retrieval filter. Polarity runs showed the lack of the classification accuracy of the polarity classifier. We need to find more descriptive features which identitfies the polarity of a sentence. Further more, complex negated statements requires a lot more processing for us to be able to find its polarity correctly.

8 Acknowledgement

This work is in conjuction with Minekey Inc. We would like to show our gratitude for allowing us to use their opinion data.

References

[1] Tim Finin Anupam Joshi Akshay Java, Pranam Kolari and Justin Martineau. Blogvox: Separating blog wheat from blog chaff. *In Proceedings of the Workshop on Analytics for Noisy Unstructured Text Data, 20th International Joint Conference on Artificial Intelligence (IJCAI-2007).*

[2] Machine Learning Library Apache Mahout. http://lucene.apache.org/mahout.

[3] Lillian Lee Bo Pang. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *In Proceedings of the ACL.*

[4] Lillian Lee Bo Pang. Thumbs up? sentiment classification using machine learning techniques. *In Proceedings of EMNLP.*

[5] Apache Lucene. Apache software foundation, http://lucene.apache.org.

[6] Craig Macdonald and Iadh Ounis. The trec blog06 collection : Creating and analysing a blog test collection. *DCS Technical Report TR-2006-224. Department of Computing Science, University of Glasgow. 2006.*