UALR at TREC: Blog Track

Hemant Joshi, Coskun Bayrak, and Xiaowei Xu
Information Retrieval Group
{hmjoshi, cxbayrak,xwxu}@ualr.edu
University of Arkansas at Little Rock

Abstract

We consider Opinion Blog retrieval from classification point of view. We used the active learning method with an integrated feature selection to train the Support Vector Machine algorithm. We wanted to study the effect of different types of features on the classification accuracy of the model generated by the classifier algorithm. We considered mainly three different types of features for 5 runs submitted. Feature types include bag-of-words features, seed-words as features and statistical features. Bag-of-words features are generated from the actual blog data. Seed-words were manually generated specific to the domain of interest. Statistical features studied included the ratio of linguistic features to total number of words. We built models using an iterative process and studied accuracy as well as coverage of each model. Study of different features is important in order to build a better model. Feature selection algorithms can choose the best features among the available ones but different features have costs associated with them. We need features that not only predict class labels or contribute towards prediction but the feature should also be representative of the entire dataset, especially test data. Training the classifier on such features will yield better coverage and training accuracy for the model. We compared the three different models generated by three different feature generation strategies. Our preliminary results indicate that seed-words that are specific to a particular domain or particular type of classification achieve better accuracy and coverage. In general, bag-of-words features are tightly coupled with the data they represent. On the other hand, statistical features are independent of the actual words used. Statistical features are more useful in building robust models that can be used with different languages and for different tasks.

1. Introduction

It is the first time that the Information Retrieval Group at the University of Arkansas at Little Rock (UALR) has participated in the blog track competition. A dataset of over 3 million blog posts (known as permalinks) was provided in raw form. Along with permalinks, data was also provided as blog feeds and homepages. The task this year was to identify opinionated blog posts for the given target. The target could be an entity, organization, event, person or location. *50* queries were given as targets and participants were asked to submit top *1000* results for each target ranked by opinionated nature of the blog post.

The dataset of over *3 million* permalink documents consists of blogs as well as non-blogs (news articles etc.). The non-english blogs as well as blogs with offensive content were considered as not-opinionated. From the total of *3,215,171* permalink documents, we removed those which are empty or non-english documents. The remaining *2,806,645* permalink documents were parsed to obtain text documents.

We divided the task of identifying opinionated blog posts about the target into two components. The first component is an indexing and searching component that

obtains all blog posts about the target sorted by their similarity score. We used normalized term frequency (*TF*) and Inverse Document Frequency (*IDF*) metric to note the weight of each unique term in the corpus. We obtained *3,691,472* unique words in *2,806,645* permalink documents. Such a high number of unique words is due to the addition of the large number of non-english and non-dictionary words. The words commonly used as the expression blog content such as *aaarrgh!* etc. are not dictionary words but they indicate the opinionated nature of the content (probably frustration in this example). We also did not use standard stop list available with SMART [1] system but rather, manually created the list of *271* words with only relevant stop words. Words like *i*, *we* are indicative of the subjective nature of the blog and hence were not discarded.

The second component of the system uses Machine Learning classification techniques to identify opinionated blog posts from not-opinionated posts. We considered using topic detection mechanisms such as the one used in [2]. Due to lack of time constraints, we assumed that the content of the entire blog post is about the given target if the word appears in the corresponding blog post. Figure 1 shows the detailed description of the two components of our system. The Opinion Miner component is responsible for training the Support Vector Machine (SVM) classifier with linear kernels. It also predicts actual test data with respective probabilities. These probabilities are considered as scores (*score-1*) for the Opinion Miner component. The higher the score, the more opinionated is the nature of the blog post. The second component is exclusively for searching relevance of the query with blog post. This component provides relevance of each blog post as a score (*score-2*) and sorts them in descending manner. Ultimately, we integrated the scores from Search Component and scores from Opinion Miner to obtain final scores (*score-1* * *score-2*) of all blogs sorted in ascending order. Top *1000* results were returned for each query submitted to the system.

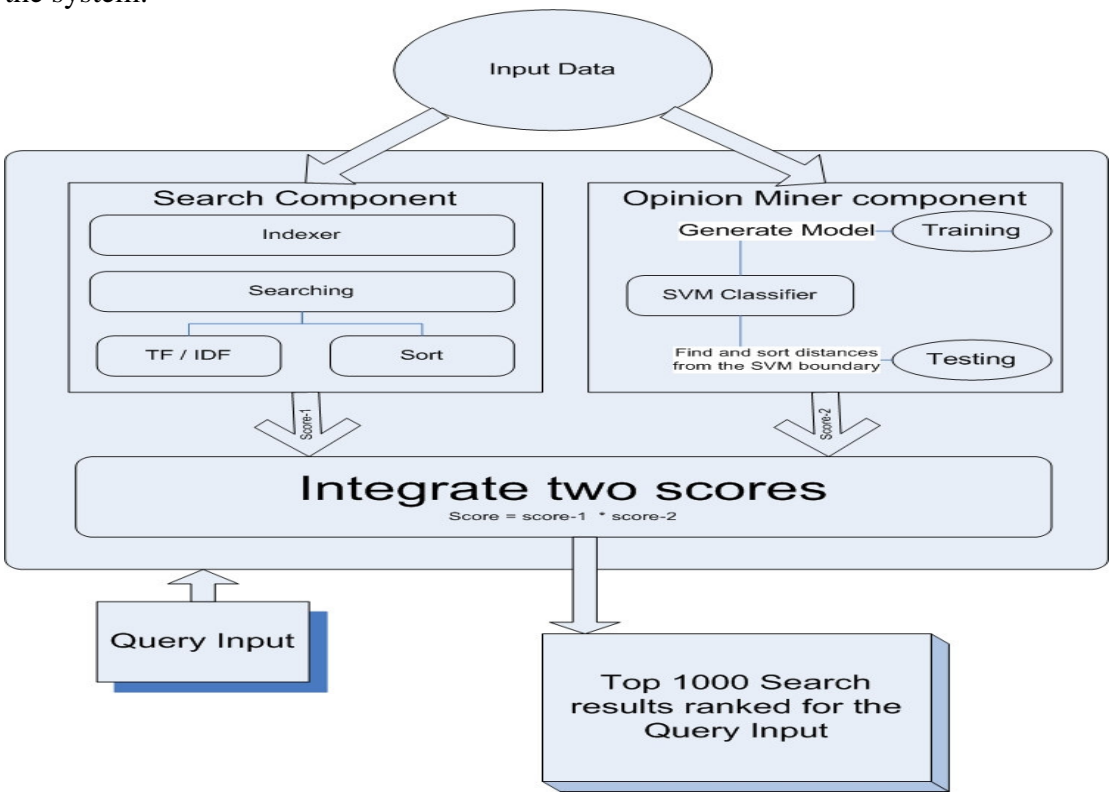

Figure 1: Overall view of the system with Search and Opinion Miner components

1.1. Search Component

We used open source *lucene* search engine [3] for indexing about *2.8 million* text files. Each text file represents the content of the blog post. *Lucene* supports inverted index structures as well as stemmers (We used Porter stemmer [4]) and analyzers to index the documents. Standard normalized *TF * IDF* weighting was implemented after filtering the stop words. The index files were merged to faster access the index (*8 GB size*). For any given query, *lucene* can compute scores of all documents being returned as relevant to the query. All returned results are sorted in an ascending order by the scores.

1.2. Opinion Miner Component

The Opinion Miner component is solely responsible for identifying opinionated blog posts. We use linear kernel SVM for training the classifier. As none of the blog posts provided were labeled, we wanted to use minimum training effort (including manual labeling of classes) and obtain maximum training accuracy. In order to accomplish this, we used the active learning paradigm [5]. We started with a small randomly selected training dataset of 20 blog posts and labeled classes as *op* or *nop* for opinionated and not-opinionated blog posts respectively. The samples were chosen such that we had 10 samples from *op* class and 10 from *nop* class. Next, we trained linear SVM with *libSVM* [6] software on the training set and built the model. Figure 2 explains active learning process in detail.

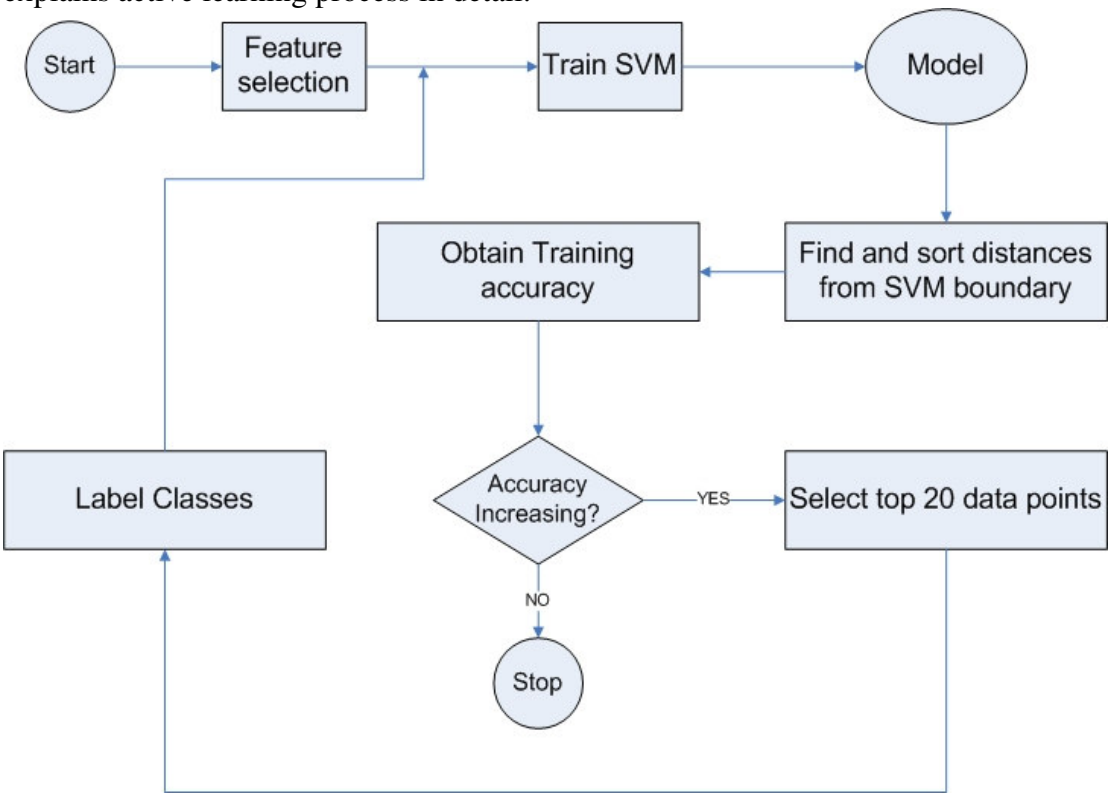

Figure2: Integrated active learning approach

We also modified the source code for *libSVM* to predict absolute distance of all test data samples from the generated decision boundary. We sorted all test data according to increasing distance from the decision boundary. This helps us identify those points that are closest to the boundary and thus most difficult for the model to predict. We selected 20 such test data points and labeled them. Now with the new training dataset of 40 samples we continued the same procedure. We noted the training accuracy until we obtained satisfactory training phase. We used *10* fold cross validation to determine training set accuracy. Adding new samples iteratively; increases the coverage of the model built using this approach. We define the coverage as the number of blog posts that can be predicted by the generated model. The higher the coverage of the model, the better it represents the actual test data.

We used active learning approach and generated different models with different set of features. We considered *3* different set of features to represent the feature space. In the first approach, we used simple bag-of-words approach to obtain all the words in the training dataset and used *Information Gain* feature selection technique to rank all features according to their values. We selected top *260* features as the best representative of the entire feature space. One of the important considerations for using feature selection was processing time. By limiting the number of features to *260*, we were able to process blogs and generate better models and coverage. We submitted automatic [*UALR06a260r2*] and manual [*UALR06m260r3*] runs obtained using Information Gain feature selection method.

We could not fully integrate feature selection as well into the active learning process due to lack of time. Using just feature selection integrated with active learning [7] does not produce expected increase in accuracy. This is because the features selected in that particular iteration ranked by *Information Gain* or any other feature selection technique may not be sufficient to truly represent the test data. Instead, integrating manual feature selection with active learning exhibited to have better results. Hence we used the second set of features totally independent of training set. We compiled a list of *500* adjectives, adverbs and few verbs that we think commonly indicate the opinionated nature of blogs. Words that indicate subjectivity or opinion are good candidates to be in this list. We used this manual seed word list as a feature set and submitted automatic [*UALR06a500r4*] as well as manual [*UALR06m500r5*] Runs. Table 1 shows the comparison of all runs submitted to the TREC.

As a third type of feature, we used simple statistical features. We counted the number of adjectives, adverbs, ratio of sum of adjectives and adverbs to total number of words for each blog post. We also added features such as ratio of the number of adjectives in the particular blog post to the highest number of adjectives found in any of the corpus blog posts etc. We used such *5* features and generated the naïve statistical model. The hypothesis for using purely frequency based or statistical features was that presence of adjectives and adverbs may be indicative of the opinion in the blog post. As can be seen from Table 1, this method did not yield very high training accuracy. We submitted only manual run [*UALR06m5Tr1*] as the fifth run for the competition.

Table 1: 5 Runs submitted to TREC: Blog Track

	Run	Type	No. of Features	Coverage	Training Accuracy*
1	UALR06m5Tr1	Manual	5	2806645	48.21%
2	UALR06a260r2	Automatic	260	2223295	74.33%
3	UALR06m260r3	Manual			
4	UALR06a500r4	Automatic	500	2711173	80.67%
5	UALR06m500r5	Manual			

2. Experiments and Results

Table 1 details the 5 runs submitted this year. Our highest priority runs were *UALR06a260r2* and *UALR06a500r4* with *260* and *500* features respectively. Figure 1 shows the interpolated precision recall response comparison for all 5 runs. It should be noted that though all runs have almost similar precision-recall response, automatic run with active learning and *260* features [*UALR06a260r2*] has a little better values than the other runs for interpolated precision at different levels of recall.

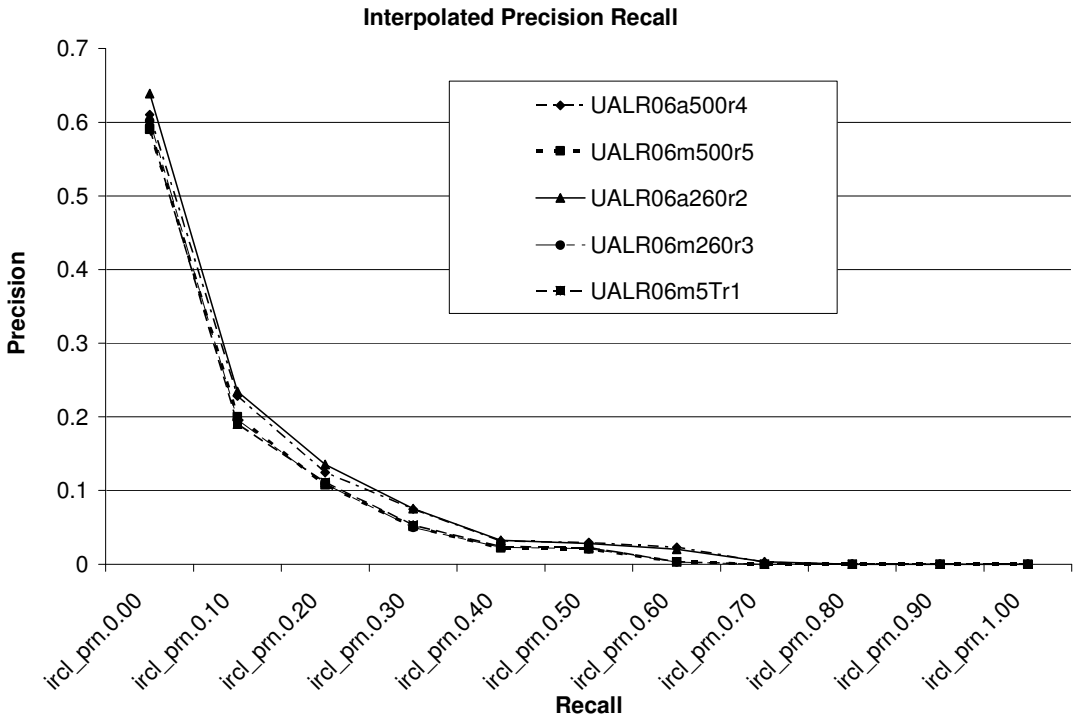

Figure 3: Interpolated Precision Recall response of the 5 runs

* 10 fold cross validation accuracy

Another comparison is shown in Figure 4. Figure 4 shows Precision response comparison for all 5 runs at different recalls. P5 indicates precision after first 5 results were retrieved and P1000 indicates precision value after first 1000 results were evaluated. Run *UALR06a260r2* shows better precision values at different recall levels.

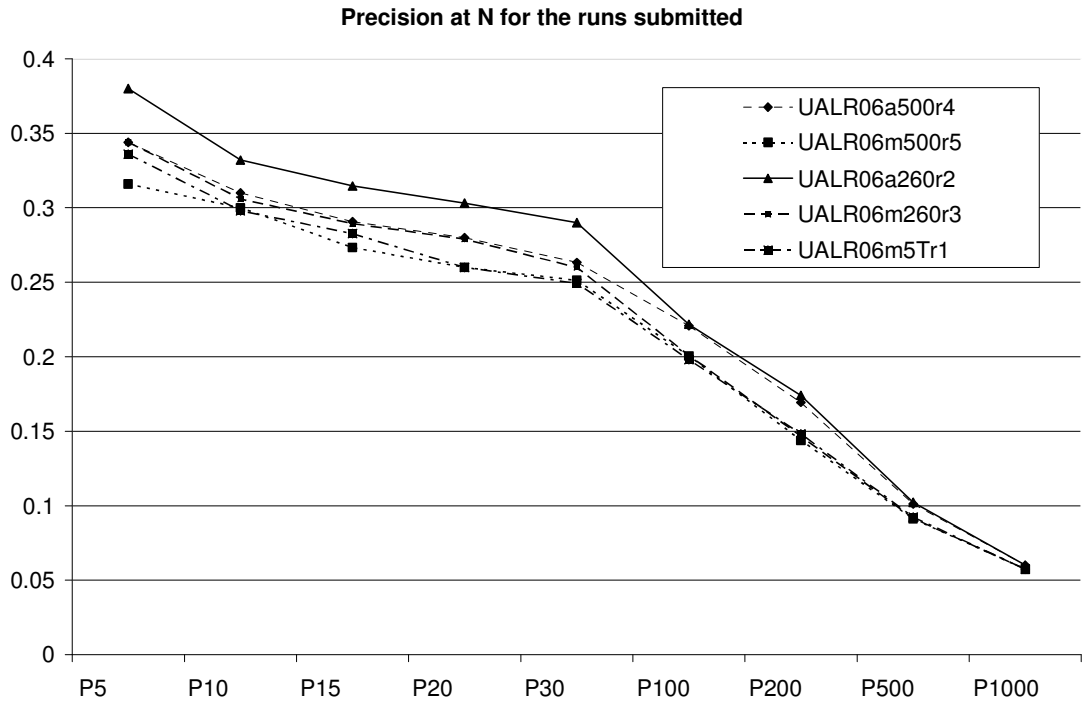

Figure 4: Precision at various levels of recall

In the future, we would like to improve active learning methods by incorporating feature selection. Preliminary work done by Raghavan et. al. [7] indicates that feature selection integrated with active learning does not always yield increasing accuracy especially with text data. We would like to investigate reasons for decrease in performance and improve the active learning algorithm.

3. References

[1] Buckley, C.: *Implementation of the smart information retrieval system*. Technical Report 85-686, Cornell University (1985)

[2] Matthew Hurst and Kamal Nigam. *Retrieving Topical Sentiments from Online Document Collections*. In Document Recognition and Retrieval XI. pp. 27--34. 2004

[3] Apache Jakarta Lucene Open source search engine available at http://lucene.apache.org

[4] Porter stemming algorithm, http://www.tartarus.org/~martin/PorterStemmer/

[5] Tong S., Koller D., *Support Vector Machine Active Learning with Applications to Text Classification*, Proceedings of ICML-00, 17th International Conference on Machine Learning, 2000

[6] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : *a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[7] Raghavan H. , Madani O. , Jones R. , *Active Learning with Feedback on Both Features and Instances* , Journal of Machine Learning Research, 2006, pp 1655-1686