BlogVox: Learning Sentiment Classifiers*

Justin Martineau, Akshay Java, Pranam Kolari, Tim Finin, and Anupam Joshi
University of Maryland, Baltimore County 1000 Hilltop Circle, Baltimore, MD 21250 410-455-1000 extension 6338
{aks1, kolari1, finin, joshi, jm1}@cs.umbc.edu
James Mayfield
Johns Hopkins University Applied Physics Laboratory
james.mayfield@jhuapl.edu

Introduction

While sentiment detection, identification, and classification are popular research areas, researchers frequently work in only one domain at a time. Typical domains include movie reviews (Pang *et al.* 2002) and product reviews (Dave *et al.* 2003).

Performing sentiment detection upon keywords chosen at run time is more difficult. The techniques applied to determine the sentiment of keywords in movie and product reviews are less effective when used on blogs due to a variety of reasons. Unlike reviews blogs tend to talk about many different subjects at a time making many NLP and machine learning approaches more difficult. Finally, many of the techniques used in the different review domains incorporate domain specific knowledge.

The 2006 NIST TREC Blog track (Ounis *et al.* 2006) on "opinion retrieval" from blog posts, presents an opportunity to tackle this problem. The task was defined as follows: build a system that will take a query string describing a topic, e.g., "March of the Penguins", and return a ranked list of blog posts that express an opinion, positive or negative, about the topic. NIST provided researchers with a data set of over three million blogs, and judged entries upon retrieval results for a set of fifty test queries.

BlogVox Design

BlogVox (Java *et al.* 2006), developed for the TREC blog track [1], performs opinion extraction upon blog posts. After data cleaning, blog posts are indexed using Lucene [2], an open-source search engine. Given a TREC query a set of relevant posts are retrieved from Lucene and sent to the scorers. BlogVox uses a meta-learning approach in order to dynamically learn topic sensitive sentiment words. In BlogVox a set of scorers individually evaluate relevant documents. These scores form a feature vector for an SVM to classify retrieved documents. A description of the SVM used is available in (Java *et al.* 2006). Figure 1 shows an overview of BlogVox.

*Partial support provided by IBM and by NSF awards ITR-IIS-0326460 and ITR-IDM-0219649.

[1] http://trec.nist.gov/tracks.html

[2] http://lucene.apache.org/java/docs/scoring.html

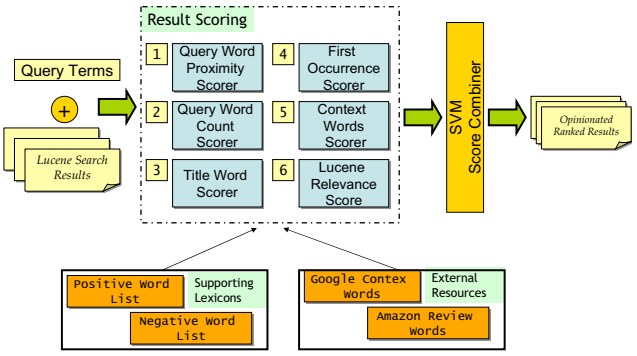

Figure 1: After relevant posts are retrieved, they are scored by various heuristics and an overall measure of opinionatedness computed by an SVM.

Most modules are simple. The query word count scorer counts query term occurrences. The title word scorer checks if query terms are present in the title. The first occurrence scorer finds the distance to the first query term. The Lucene relevance score is how closely the post matches the query terms.

Other modules are more complex. The query word proximity scorer finds the average number of sentiment terms occurring within the *vicinity* of a query term. Sentiment terms around phrasal queries recieve a boosted score (approximately twice). Sentiment terms come from either handcrafted lists or from context learners. The context word scorers find contextual terms pertinent to the topic or query using web search results as external sources.

Context Learning

The meaning and connotation of a word depend upon context. Take, for example, a car with a smooth ride, and smooth tires. Normally, some words do not express sentiment, but in context they can. However, enumerating the sentiment of words given their domain is inadvisable. In single domain problems it might be, but when queries are chosen ad hoc from a wide array of choices complexity rises too quickly.

BlogVox uses multiple different context word scorers to detect and evaluate context specific sentiment bearing terms.

Different methods should pick up on a different variety of context words.

The first module uses the Google API to obtain web documents where the original query has been sprinkled with generic positive and negative sentiment words such as "hate", "love", "sux", "annoyed", "great" to bias results to sentiment bearing pages. Relevant terms are mined from the summaries of these pages based upon TFIDF with our database of blogs. After mining, blog posts are scored based upon the number of relevant terms found. The second module uses keywords from reviews in Amazon product categories instead of terms mined from Google page summaries.

Similarly, the third and fourth modules retrieve from Google three sets of documents based upon the original query. Generic positive terms are added to the first query, generic negative terms are added to the second query, and the third query is left unchanged. In the third module the positive and negative data sets are compared against each other. On average, neutral terms should occur equally in both data sets and opinionated terms should occur more often in one set than another. Therefore, sentimentality increases with distance from the average. Following this heuristic the base opinion scores for context words are generated. In the fourth scorer the positive and negative sets are compared against the neutral set. Words with a higher probability of belonging to an opinionated data set receive a higher opinion score.

In modules three and four, blog posts were scored by summing the opinion scores for all words, weighted by inverse distance to a query term. Since finding the right distance weight function was difficult, we used SVMs. The count of all words with a base opinion score within a set range, and with the nearest query term also within a set range form the features of the SVM. Tested on a data set with queries ranging from products, to places, to people, to movies and tv shows this method was around 70% accurate using 10 fold cross-validation.

Evaluation and Future Work

BlogVox measures search results both by how relevant they are and how opinionated they are. For opinion retrieval MAP (Mean Average Precision) was 0.0764 and R-Prec (Precision after retrieving R documents, where R is the number of relavant documents) was 0.1307. For topic relevance MAP was 0.1288, and R-Prec was 0.1805. Our Scores are around the median scores across all submissions. Figure 2 shows our TREC results in blue for opinion.

Web documents are a promising source for domain specific learning, however speed and document quality are issues. Using the full content of web documents, to discover sentiment words, rather than their summaries, trades speed for quality. Faster learning algorithms can alleviate this problem.

We are working on mitigating noise in blog posts caused by splogs, spurious post content e.g., blogrolls, advertisements, sidebars, navigation panels, headers and footers. See (Java *et al.* 2007) for further information about noise reduction techniques. Since advertisements tend to be positive, adapting these techniques to regular web pages would improve results.

In addition to enhancing old scorers we are expanding BlogVox with new scorers such as an adjective word count scorer. This scorer uses an NLP tool to extract adjectives around the query terms. However, the noisy and ungrammatical sentences present in blogs impacts its performance.

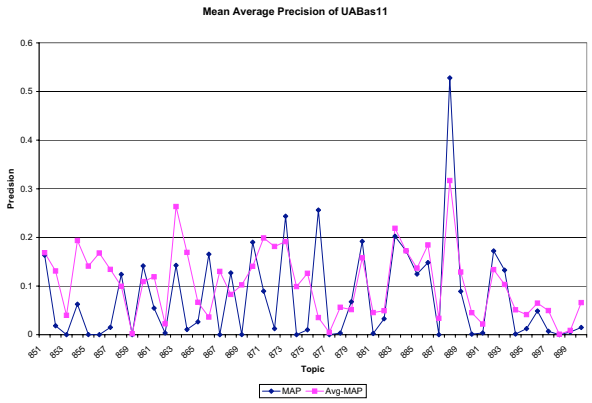

Figure 2: Mean average precision of submission UABas11

Conclusion

Performing sentiment analysis upon a topic, specified by key words, without prior knowledge about the key words is a difficult task. With the growth of the blogosphere researchers, corporations, and politicians, among others are very interested in applying sentiment detection to blogs. To accommodate the demands from myriad users, with similarly diverse desires, a sentiment analysis engine for blogs must discover domain specific features relevant to queries in order to accurately assess the sentiment of blogs. Using meta-learning upon the results of web searches, as BlogVox does, can accomplish this goal.[3]

References

K. Dave, S. Lawrence, and D. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW*, pages 519–528, 2003.

A. Java, P. Kolari, T. Finin, J. Mayfield, A. Joshi, and J. Martineau. BlogVox: Separating Blog Wheat from Blog Chaff. In IJCAI, January 2007.

A. Java, P. Kolari, T. Finin, J. Mayfield, A. Joshi, and J. Martineau. The UMBC/JHU blogvox system. In *Proc. 15th Text Retrieval Conf.*, November 2006.

I. Ounis, M. de Rijke, C. Macdonald, G. Mishne, I. Soboroff. Overview of the TREC-2006 Blog Track. In *Proc. 15th Text Retrieval Conf.*, November 2006.

B. Pang, L.n Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proc. EMNLP 2002*, 2002.

[3] See http://userpages.umbc.edu/ jm1/BlogVox/ for supplemental work