NORTHWESTERN
UNIVERSITY

Electrical Engineering and Computer Science Department

Technical Report
NWU-EECS-07-05
July 21, 2007

Reasoning Through Search: A Novel Approach to Sentiment Classification

Sanjay Sood, Sara Owsley, Kristian J. Hammond, Larry Birnbaum

Abstract

We introduce a novel approach to sentiment classification. Our Reasoning Through Search (RTS) technique uses existing labeled data, query formation strategies, and a case base to estimate the sentiment of a text review. Unlike previous systems, when classifying a document from a domain that the system does not have explicit, in-domain training data, our classification system leverages domain relatedness and a case base of labeled reviews to perform the classification. While the system does require labeled training data, it does not rely on the guaranteed presence of in-domain labeled training data.

Reasoning Through Search: A Novel Approach to Sentiment Classification

Sanjay Sood
Intelligent Information Laboratory
Northwestern University
2133 Sheridan Road
Evanston, IL 60208
sood@cs.northwestern.edu

Sara Owsley
Intelligent Information Laboratory
Northwestern University
2133 Sheridan Road
Evanston, IL 60208
sowsley@cs.northwestern.edu

Kristian J Hammond
Intelligent Information Laboratory
Northwestern University
2133 Sheridan Road
Evanston, IL 60208
hammond@infolab.northwestern.edu

Larry Birnbaum
Intelligent Information Laboratory
Northwestern University
2133 Sheridan Road
Evanston, IL 60208
birnbaum@infolab.northwestern.edu

ABSTRACT

We introduce a novel approach to sentiment classification. Our *Reasoning Through Search* (RTS) technique uses existing labeled data, query formation strategies, and a case base to estimate the sentiment of a text review. Unlike previous systems, when classifying a document from a domain that the system does not have explicit, in-domain training data, our classification system leverages domain relatedness and a case base of labeled reviews to perform the classification. While the system does require labeled training data, it does not rely on the guaranteed presence of in-domain labeled training data.

1. INTRODUCTION

People have opinions about things they encounter in their day to day lives such as products and services. Traditionally, companies have captured such opinions through customer satisfaction surveys and focus groups in order to understand their users' needs and improve products and services to meet these demands. The emergence of vast amounts of opinions online in the form of professional product reviews, consumer generated product reviews, newsgroups, weblogs, and news articles has given rise to an opportunity to collect this information on a large scale, without explicitly requesting it from a consumer.

In order for these information sources to be useful for marketing and business intelligence on a large scale [10, 12], companies must be able to automatically classify the author's age, gender, geographic location and other demographic information as well as the sentiment of their opinion on a product. Towards these goals, this paper focuses on the task of classifying the sentiment of a text, specifically focusing on product and service reviews.

WWW2007, May 8–12, 2007, Banff, Canada.

Our goal was to build a system that could classify the sentiment of reviews across disparate domains without guaranteeing the presence of any in-domain labeled training data. Classifying sentiment of a document without knowing its domain and without sufficient training data is particularly hard given the inconsistencies in the connotation of sentiment words across domains. A word might have a very positive meaning in one domain, but may have a negative meaning in another domain. For example, a cell phone described as 'small' may be considered desirable, but when applied to a television, smallness is not usually considered a coveted trait. Conversely, domains may share language to convey sentiment. For example, people often use many of the same words to describe what they liked and disliked about movies and books.

We introduce a novel approach to sentiment classification. Our technique uses existing labeled data, query formation strategies, and a case base to estimate the sentiment of a text review. Unlike previous systems, when classifying a document from a domain that the system does not have explicit, in-domain training data, our classification system leverages domain relatedness and a case base of other labeled reviews to perform the classification. While the system does require labeled training data, it does not rely on the guaranteed presence of in-domain labeled training data. Until all text on the Web is clearly tagged and categorized, systems that can work across multiple domains, trained or untrained, will be critical to leverage the large corpus of data on the Web.

2. PREVIOUS WORK

Much of the previous work in sentiment classification [1, 23, 22, 18] has dealt with single domain classification where there are large amounts of labeled data available for training. A large portion of this work has been concerned with getting the best in-domain classification accuracy by using various machine learning strategies (Naïve Bayes, SVM, Maximum Entropy, etc.) and selecting the most appropriate feature set

(unigrams, n-grams, adjectives, etc.) to train from. From our investigation, little work has been done in trying to create a sentiment classifier that can operate across new domains without labeled training examples.

Other researchers have approached this problem without considering domain specificity. Nasukawa et al [15] built a system for extracting sentiment at a sentence level, extracting polarity for subjects in a document. Another similar approach from Popescu and Etzioni [19] uses an unsupervised method (relaxation labeling) to extract and analyze opinion phrases corresponding to features as opposed to classifying the entire document. Both of these lexogrammatical approaches have good performance, but rely on manual coding of words and their polarity or lexico-syntactic patterns.

While building a system to classify subjectivity/objectivity and sentiment (positive/negative), Finn and Kushmerick [9] noted that the performance of their classifiers degraded significantly when applied to a new domain. To build a classifier that worked across domains, they used an ensemble approach with different feature sets, but the system's performance peaked at 50%. Other work [17] in sentiment classification has shown clear differences in the polarity of words from a domain-specific classifier and a general purpose affective corpus. A need was expressed for domain-specific classifiers in order to appropriately classify the sentiment of text from different domains, since the differences in meaning of words between domains would reduce the accuracy of classification.

A domain-specific approach, however, requires training data in every domain that would be encountered in classification. Aue and Gamon surveyed various techniques for sentiment classification in new domains and concluded that labeled training examples from the new domain are needed for accurate classification [3]. Enumerating and finding the appropriate labeled data in every single domain is time-consuming and not scalable to all the domains one my encounter on the Web. There is also the problem of determining the correct classifier to apply for text where the domain is not known beforehand. This problem is apparent in attempting to classify weblogs as they are unstructured, unlabeled, and unedited. While topic classification [16, 13] has traditionally had better performance than sentiment classification, it also requires having training data for every topic the system will consider.

Approaching text classification from a different angle, systems have been built that use case-based reasoning to classify text such as e-mail spam – a task often approached using machine learning techniques. Cunningham, et al, argue that case-based classification works well when there is a variation among individual cases that cannot be captured in a high level statistical model [7, 5, 11]. Similarly, given the variation in the affective connotation of language when used across domains [17], we feel that a case-based approach to sentiment classification could yield better accuracy by leveraging individual cases as opposed to a unified statistical representation.

3. REASONING THROUGH SEARCH (RTS)

The system we propose uses a combination of machine learning, information retrieval techniques, and a case base to determine the sentiment of a review published on the Web.

We interpreted sentiment classification as being a binary classification problem between positive and negative. Figure 1 shows an architecture diagram of the RTS system. We begin by transforming text into a set of features. Instead of using a probabilistic model such as Naïve Bayes to classify the set of features as positive or negative, we leverage a statistical model of training data to create a representation of the target document, in the form of a sentiment query. This query is used to retrieve singular cases of labeled data from a case base. In addition, we use the feature representation of the document to retrieve a closest-fit ranked list of known domains. The labels of the returned cases and the ranked domain list are processed by a case evaluator to extract a score for the sentiment of the document.

3.1 Data

We gathered our data from Rate-It-All [21], an online repository of consumer written reviews on a wide variety of topics including: products, services, famous people, sports teams, movies, music, and colleges. The reviews each have an associated rating, 1 to 5 stars, assigned by the author. Once submitted to RateItAll, the reviews do not go through an editorial process and are presented as is. We chose a subset of domains from RateItAll that we felt covered a breadth of topics. The domains we selected were: actors, books, colleges, destinations, drinks, electronics, food, movies, music, restaurants, software, sports and video games.

We collected a total of 106,961 reviews from these 14 domains. Some reviews consisted of a star rating with no review text or a very short review text, so we limited our collection to reviews with 6 or more words. The average length of a review was 47.86 words, with a minimum length of 6 words and a maximum length of 1205 words. Given that the reviews were rated between 1 and 5, we labeled the set of negative reviews as those with 1 or 2 stars, and positive reviews were those with 4 or 5 stars. Reviews with 3 stars were ignored as they were seen to be neutral.

While this corpus of reviews is very useful as a training corpus, it does have some anomalies. Since the reviews do not go through any editorial process, they often contain misspellings, use slang words, and are off topic. Reviewers occasionally make mistakes in terms of the number of stars they assign to a review. Since RateItAll exists as a social network as well, the reviews often contain dialog between reviewers. As with any free text, human-generated content, such noise is unavoidable.

3.2 Domain Classifier

Previous work in sentiment analysis established, as discussed previously, that words have different affective connotations across domains. Since our case base contains reviews across all fourteen domains, retrieving relevant cases requires knowledge of the domain(s) that the current document is similar to. To meet this need, we built a Naïve Bayes domain classifier because of the relative accuracy and ease of implementation. Given a document from an unknown domain, the classifier returns a list of domains, ranked and weighted from most related to least related, where the weights are normalized probabilities.

The training data for this classifier was reviews across fourteen domains, described in the previous section. To train the classifier, we treated each unigram as a feature of a document, while employing Porter's stemmer [20] to compress terms with morphological variation. While pre-

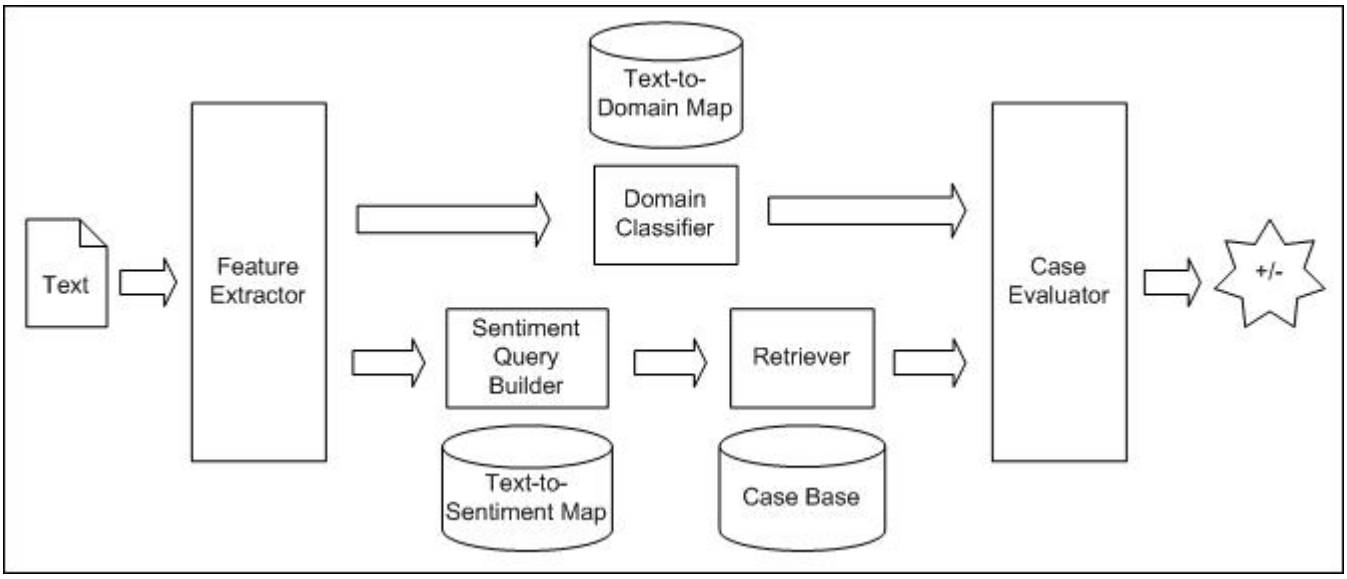

Figure 1: An architecture diagram of the RTS system.

vious work has shown that alternative feature extraction (bigrams, adjectives, phrases, etc.) provides an improvement in performance [4, 14], we found the performance of a system using unigrams as features to suffice for our purposes. Each training document, d, was split into a vector containing the n unique features that appeared in the document, capturing the presence of a feature in a document and not the frequency [18] of the features in the document: $d_i =< f_1, f_2, f_3, ..., f_n >$.

Combining these feature vectors with the known domains of the training documents, the probabilities for each feature in the classifier were calculated. For flexibility, we created the domain classifier by building a probabilistic sentiment model (positive/negative) for each domain. This allowed us to generate a Naïve Bayes statistical model on the fly for both domain and sentiment classification.

Given the target document, a ranked and weighted list of domains is created by calculating the NB probability that the document d is a part of each candidate class c, where c is one of the domains. The standard NB classifier employs a product of the probabilities, however, to prevent underflow of the product, we used the summation of the logs of the probabilities. We also used add one smoothing to prevent the length of the document or the amount of training data in each domain from influencing the classification. The following equation was used to get the NB probability that the document d is a part of each class c:

$$P_{NB}(c|d) = P(c) + (\sum_{i=1}^{m} \log(P(f_i|c)^{n_i(d)}))$$

3.2.1 Domain Relatedness

After we created our domain-specific sentiment classifiers, we experimented with sentiment classification within and across domains. The goal of this experiment was to verify the notion that sentiment classifiers are domain-specific and do not work across domains.

For testing we took each domain classifier, 14 in total, and tested it's accuracy in classifying sentiment in every domain, including its native domain. In testing on the native domain, we employed 5-fold cross-validation. Table 1 shows a selection of results from the experiment. The data showed that the highest accuracy for each domain classifier was when it was tested against the native domain data, which is expected since it is in-domain classification. Based on classification accuracy, we observed distinct clusters between domains that are topically related. For example, the classifier for 'restaurants' performed well over testing data from the 'food' domain and vice versa. There was poor performance with some pairs of domains. For example, the accuracy of the 'movie' classifier when applied to 'colleges' was only 41.5%, below the 50% baseline accuracy for binary classification. To explain this result, we theorize that there is disagreement among discriminating sentiment terms between the two domains – some terms that are positive in movies have a negative connotation when applied to colleges.

These results show that there are definite relationships between domains based on common language usage. It seems reasonable to believe that people would use similar language to describe books and movies. It also seems reasonable to agree that people talk about cars and music differently. With this in mind, domain relationships and individual cases in other domains can be leveraged for classifying sentiment.

3.3 Case Retrieval

The next step in the sentiment classification process involves creating a representation of the target document that extracts features related to sentiment. This representation can then be used to retrieve similar cases from a case base.

3.3.1 Query Formation

Transforming a document to a set of term based queries is a well-known method for information systems that provide relevant, related content. Systems, such as Watson [6], use the document as a starting point for finding similar content

Table 1: A sample of Naïve Bayes sentiment classification accuracies within and across domains.

Classifier	Testing Domain	Accuracy
actors	actors	76.99%
colleges	colleges	82.11%
food	restaurants	77.14%
food	software	41.90%
movies	colleges	41.54%
restaurants	destinations	70.29%
software	electronics	68.00%
sports	music	41.61%
video games	software	68.04%

by forming a term-based query and retrieving information from disparate information sources. In Watson, the model of relevance is almost always viewed as a function of topical similarity. In our system, however, we viewed the problem as being that of finding results based on affective similarity – using features that are highly discriminating for sentiment as the query-based representation of the document.

During the training phase, we calculated a probabilistic model for sentiment (positive/negative) over each domain, giving us sentiment information for a given unigram across all domains. We used this model to extract the strongest affective unigrams from the target document. To do this, we tokenized and stemmed the target document. For each word, we calculated the sentiment magnitude of the word w using the following formula:

$$sm(w) = \sum_{i=1}^{m} abs(\log(P_i(w|pos)) - \log(P_i(w|neg)))$$

where i denotes a domain in which we are examining the probabilities, with m domains in total.

Words with high sentiment magnitude are seen as being discriminating terms for sentiment. While this does not provide an overall sentiment (positive/negative) for a word, since we are taking the absolute value, it considers that a word might have a positive overall connotation in one domain, but a negative in others. The resulting score measures the overall, absolute polarity of a word across all domains.

Once we calculated a sentiment magnitude for each term in the review, we formed a term based query by sorting the resulting vector by sentiment magnitude. For each candidate word we scanned a 3 term window around the term in the document looking for modifiers (very, not, too, etc). In the case of finding a modifier within the window, the term was expanded to the phrase that includes the modifier. For example, the term 'funny' with a high sentiment magnitude would be transformed to 'not funny' to reflect its meaning in the target document.

After completing expansion, the system created a term-based query by concatenating the terms with the highest calculated sentiment magnitude. We limited the length of each query to be 4 terms, not including term expansion with modifiers. The generated query is then used to retrieve related cases from our case base.

In some very rare cases a generated query did not return results because the representation created for the document did not match any cases in the case base. In these instances, the query was relaxed and resubmitted to the case base by removing terms from the right of the query vector. Removal of words creates a more general query that has a higher likelihood of matching other cases.

3.3.2 The Case Base

For case retrieval, we wanted to be able to retrieve labeled reviews by text-based similarity to a generated query. Instead of implementing our own retrieval system, an off-the-shelf search engine was sufficient for our purposes. We employed Apache Lucene [2], an open-source search engine, to index and retrieve cases. All labeled reviews described above in the data section were indexed in the engine.

The standard Lucene setup was used, indexing both the content of the reviews, as well as the known domain of the review. This allowed us to retrieve cases based on textual similarity to the review and to filter based on the domain of the review. The Lucene engine was originally setup to ignore stopwords during indexing. This meant that searching for the phrase "United States of America" would yield no results, though the phrase did exist in the case base, because the word 'of' was not included in the engine's index. We modified the engine so that all words, including stopwords, were indexed, allowing us to include phrases, such as those described in the previous section, in our queries.

Each review's file name was created such that it contained meta-data including its domain, the rating assigned by its author (one to five stars), and a unique identifier (e.g. music-4-8273.txt). The body of the text file contained the text of the review.

3.4 Result Ranking and Document Scoring

After a set of affect-similar cases have been returned from the case base, a sentiment score is calculated for the target document based on the sentiment scores attached to the retrieved cases. We only looked at the top 25 results returned from Lucene when calculating a sentiment score. Given that a target document may be closely related topically to a known domain in the system, the ranked and weighted domain list generated in domain classification is used to weight each sentiment score for the returned cases. The overall score for the document is calculated as follows.

Given that *ls(c)* returns the labeled score of a case c and *w(c)* returns the weight of the labeled domain of a case c.

$$sc(d) = \frac{\sum_{i=1}^{m}(ls(case_i) - 3) * w(case_i)}{m}$$

returns the sentiment score of a document d based on m retrieved cases.

In addition to weighing the case scores from the domain classifier, we also experimented with weighing the scores by

the ranked position of the domain in the classification results. Preliminary experiments show that weighing the results using either strategy had little effect on the final classification accuracy. We speculate this has to do with the overall performance of the domain classifier, which is not phenomenal. We theorize that improving the classification accuracy of the domain classifier by using more sophisticated feature extraction techniques or another system may improve this portion of the system.

4. RESULTS

To evaluate the results from our system, we chose to compare the performance the system against other computational methods for sentiment classification of documents in unknown domains. In addition, we ran a small human study to get a sense of how well humans can classify the sentiment of reviews.

4.1 Human Study

We ran a small study that asked participants to perform such classification. To generate a questionnaire, we randomly selected the text of a set of reviews, across all domains and with a positive or negative rating, from the corpus collected from RateItAll. We created a questionnaire that contained the text of each review and asked study participants to rate each review as being positive or negative. The star score assigned to each review by the author was treated as the truth value, where a score less than 3 stars was considered negative, greater than 3 stars positive.

The study consisted of 13 participants. After dropping the highest and lowest score, the results of the study showed that assigning sentiment values to text is a non-trivial task for humans. The average accuracy of human classification was 15.72 out of 20 (78.6%) with a standard deviation of 1.55.

In aggregate, humans tended to have problems classifying short reviews that lacked sufficient context to determine the object of the review. The participants also had problems classifying reviews that used sarcasm. It is specifically these types of reviews that have been some of the major failure cases for almost every sentiment classification system to date.

4.2 The All Data Approach

One approach to text classification is to train one classifier on the labeled data in all domains [3]. In this approach, an equal amount of data from each domain is used to train a general purpose classifier. With this approach, however, variation of distinguishing features between classes over the domain set can severely degrade classification accuracy. In other words, if the affective connotation of a word varies across domains, combining the domains into one classifier will diminish classification accuracy.

To see how an all data classifier compared to our system, we used our existing infrastructure to train separate sentiment classifiers for each domain using an equal amount of data from each domain. Separate classifiers allowed us to combine the results across classifiers at run time to create an all data classifier. Once again, we used stemmed unigrams as the feature set. We then took each domain and performed sentiment classification on all the reviews using an all data classifier made up of the other 13 domains. Table 2 illustrates the classification accuracy of this approach.

Table 2: Average accuracies of sentiment classification across all domains.

Approach	Accuracy
human baseline	78.60%
all data	66.00%
ensemble	60.66%
RTS (sim queries)	62.72%
RTS	73.39%

4.3 Ensemble Approach

Another approach to classifying sentiment in unknown domains is using an ensemble of classifiers [8]. With this method, a classifier is trained on each domain. Instead of being combined into one super classifier, as with all data, the result of each classifier is combined to produce an overall classification.

Like the all data system, the ensemble approach leverages out-of-domain labeled data to classify data in a new domain. As with any classification system, implementation details vary. Once again, we used the existing domain classifiers, described above, to provide the ensemble of classifiers.

To test the ensemble system, we performed classification for each domain against an ensemble of classifiers that included every domain except the target domain. While there are various ways of combining the results from classifiers, we set up a simple voting mechanism where each ensemble could vote whether the target document was positive or negative. Since there were an odd number of domains (when the target domain is removed) and all domain classifiers had an equal vote, classification of the text was either positive or negative, with no unknowns. The accuracy of the ensemble classifier is illustrated in Table 2.

4.4 Our System

To evaluate our system, we ran the system across each of the fourteen domains, measuring classification accuracy. Since we are interested in the performance of classifying unknown domains, we removed all in-domain training data at runtime. This includes removing training data for the target domain from the domain classifier, sentiment term extractor, and the case base. Removing this domain-specific information functionally made the target domain unseen in the system, since there was no in-domain labeled data present in any of the processing steps.

The accuracy of the RTS system compared to other approaches is illustrated in Table 2. The average accuracy of sentiment classification across all domains was 73.39%. More detailed results showing classification accuracies across each domain can be seen in Table 3.

In order to examine the performance of the sentiment queries, we decided to modify the query formation. Instead of creating a representation of the document based on the strength of terms based on sentiment, we created a query based on frequency of the terms (removing stopwords). This method gave us a similarity query that represent topically centered terms in the text. We then used that query to retrieve cases from the case base. The results of this experiment can be found in Table 2 under RTS (sim queries). The comparison between RTS, using sentiment queries, and

Table 3: The accuracy of sentiment classification across each domain using the RTS approach.

Domain	Accuracy
actors	75.50%
books	72.81%
cars	74.36%
colleges	73.20%
destinations	75.26%
drinks	73.75%
electronics	68.47%
food	67.82%
movies	73.17%
music	76.59%
restaurants	75.98%
software	70.65%
sports	68.53%
video games	68.38%

RTS using similarity queries, shows that the query formation algorithms are effective in extracting a term-based representation of the sentiment of a document.

The results in Table 2 show that our system outperformed other techniques for sentiment classification in new domains. Unlike previous results [3] in unknown domain sentiment classification, our system does not require labeled examples in the new domain. While the system did not outperform the human baseline, future enhancements to the system have the potential to close the gap.

5. FUTURE WORK

While our system has promising results, we see several opportunities for improvement. The current system uses unigrams as features for both the domain classifier as well as the sentiment term extraction algorithm. A future system could employ an information gain feature extraction algorithm and other features such as bigrams.

A known problem of sentiment classification is detecting sarcasm and irony in text. We see an opportunity to make advances in addressing this problem. In query formation, our current system extracts terms from a document that are highly discriminating between positive and negative cases using a probabilistic model of the training documents. Using this model, we could identify cases with conflicting language and employ an alternate classification strategy.

An important aspect of our system is the selection of a set of domains. We subjectively selected domains that we felt covered a large breadth of topic areas at the right level of specificity. A computational approach to domain selection is possible given the relationship between domains illustrated in Table 1.

Using human categorization as a starting point, we can validate the relatedness of documents in a domain by building sentiment classifiers within each domain and performing classification within and across domains. Given the results, domains can be combined if classification is accurate across those domains, indicating that these domains have similar language connotation for sentiment. Conversely, a domain can be decomposed if classification accuracy is poor within the domain, meaning that sentiment bearing words cannot be generalized within the predefined domain label.

6. CONCLUSION

While marketing research is an area which demands sentiment classification tools, there are other areas in which specific aspects of this system could be useful.

We have applied this system to analyze the emotional content of weblogs (blogs). We use this analysis in two current systems called *Buzz* and *News at Seven*. *Buzz* is a digital theater installation deployed as a group of virtual actors, animated faces with computer voice generation, who present compelling and emotional stories found in blogs. This emotional classification system has been critical to the success of *Buzz*, improving it's performance greatly by allowing the system to filter out the unemotional stories.

The *Reasoning Through Search* emotional classification system has also proven to be critical in *News at Seven*, a completely automatically generated news show using content found in various web resources and presented through characters in a modern game engine. For each story in *News at Seven*, it presents a blogger's point of view on the topics in the story. This system enables *News at Seven* to find opinions that are emotional and pointed.

The extraction of sentiment bearing terms from a document can facilitate the building of new search systems that index documents based on their sentiment. Coupled with existing search technology, systems can be created to allow users to browse documents by their emotional and topical content. This technology, for example, could allow users to retrieve reviews with opposing views on a product or service.

Another application for this technology is the improvement of the prosody of speech generation systems. Knowing the emotionally charged terms in a sentence can aide in generating believable and engaging speech. We are currently integrating the extraction of sentiment terms into a speech generation system tied to an avatar built to host a full stage improvisational show.

While the performance of our system does not exceed that of a domain-specific sentiment classifier, it outperforms other methods of sentiment classification over unknown domains. Our system proves to be a promising and novel approach to sentiment classification of text. This approach is particularly powerful when an explicit domain set and corresponding training data are not available – a prime example being classification on the Web.

7. REFERENCES

[1] C. O. Alm, D. Roth, and R. Sproat. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of HLT/EMNLP*, 2005.

[2] Apache Lucene. http://lucene.apache.org/, 2006.

[3] A. Aue and M. Gamon. Customizing sentiment classifiers to new domains: a case study. In *Proceedings of RANLP*, 2005.

[4] R. Bekkerman and J. Allan. Using bigrams in text categorization. Technical Report IR-408, Department of Computer Science, University of Massachusetts, Amherst, MA, 2004.

[5] S. Brüninghaus and K. D. Ashley. How machine learning can be beneficial for textual case-based reasoning. In *Proceedings of the AAAI-98/ICML-98*

Workshop on Learning for Text Categorization, pages 71–74, 1998.

[6] J. Budzik. *Information Access in Context: Experiences with the Watson System*. PhD thesis, Northwestern University, June 2003.

[7] P. Cunningham, N. Nowlan, S. J. Delany, and M. Haahr. A case-based approach to spam filtering that can track concept drift. In *Proceedings of the ICCBR Workshop on Long-Lived CBR Systems*, 2003.

[8] T. G. Dietterich. Machine learning research: Four current directions. *AI Magazine*, 18:97–136, 1997.

[9] A. Finn and N. Kushmerick. Learning to classify documents according to genre. In *IJCAI-03 Workshop on Computational Approaches to Style Analysis and Synthesis*, 2003.

[10] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo. Analyzing online discussion for marketing intelligence. In *Proceedings of the 14th International Conference on the World Wide Web*, pages 1172–1173, Chiba, Japan, 2005.

[11] M. Healy, S. Delany, and A. Zamolotskikh. An assessment of case-based reasoning for short text message classification. In N. Creaney, editor, *Procs. of 16th Irish Conference on Artificial Intelligence and Cognitive Science, (AICS-05)*, pages 257–266, 2005.

[12] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth international conference on Knowledge Discovery and Data Mining*, pages 168–177, Seattle, WA, 2004.

[13] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*, pages 137–142, 1998.

[14] S. Muresan. Combining linguistic and machine learning techniques for email. In *Annual Meeting of the ACL, Proceedings of the workshop on Computational Natural Language Learning*, volume 7, pages 1–8, 2001.

[15] T. Nasukawa and J. Yi. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd International Conference on Knowledge Capture*, 2003.

[16] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proceedings of the IJCAI Workshop on Machine Learning for Information Filtering.*, 1999.

[17] S. Owsley, S. Sood, and K. J. Hammond. Domain specific affective classification of documents. In *Proceedings of the AAAI Spring Symposium on Computational Analysis of Weblogs.*, pages 181–183, 2006.

[18] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86, 2002.

[19] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of HLT-EMNLP 2005*, 2005.

[20] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[21] Rate-It-All: The Opinion Network. http://www.rateitall.com/, 2006.

[22] P. D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *ACL*, pages 417–424, 2002.

[23] P. D. Turney and M. L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346, 2003.