Blogvox2: A Modular Domain Independent Sentiment Analysis System

by

Sandeep Balijepalli

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
2007

ABSTRACT

Title of Thesis: Blogvox2: A Modular Domain Independent Sentiment Analysis System.

Sandeep Balijepalli, Masters of Science, 2007

Thesis directed by: Dr. Tim Finin, Professor
Department of Computer Science and
Electrical Engineering

Bloggers make a huge impact on society by representing and influencing the people. Blogging by nature is about expressing and listening to opinion. Good sentiment detection tools, for blogs and other social media, tailored to politics can be a useful tool for today's society. With the elections around the corner, political blogs are vital to exerting and keeping political influence over society. Currently, no sentiment analysis framework that is tailored to Political Blogs exist. Hence, a modular framework built with replicable modules for the analysis of sentiment in blogs tailored to political blogs is thus justified.

I propose Blogvox2, an information retrieval based domain independent sentiment analysis framework that uses customized pattern matching techniques, such as nave bayesian filter, bag of words and part of speech tagging are used for opinion extraction in blogs. We also developed prototype two-panel and four-panel search applications of the query results. In addition, we also analyze opinionated sentences to identify trends on the hot and top topics.

The modular framework of of Blogvox2 provides a platform where new modules for different domains can be easily plugged in. The framework provides the date of publishing, permanent link and the urls of the sentences that expresses opinions based on the analysis.

Based on the analysis of the blogvox2 on political domain, our system performs well with unigram approach. We investigated our framework with pattern matching techniques, bigram technique, combining the unigram and bigram techiniques and incorporating parts

of speech tagging, which have not fared as well as unigram techniques. We also investigated the reasons for the performance degradation or enhancements on each approach. Based on our analysis, we developed different applications to ease the use of our framework.

Dedicated to Ma, Nana and Sridhar

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my graduate advisor Dr. Tim Finin. His suggestions, motivation and advice have proved very valuable in this work.

I take this opportunity to thank Dr. Anupam Joshi and Dr. Yelena Yesha for graciously agreeing to be on my thesis committee. Dr. Finin and Dr. Joshi were instrumental in providing new ideas, pointers to related work and timely feedback on all aspects of my work.

I thank Dr.Lada A. Adamic as well as companies such as Tailrank, Twitter for allowing me to user their respective datasets for experimental validations.

I would also like to thank Akshay Java and Pranam Kolari for their support, valuable discussions and time. I am specially obliged to Justin Martineau for his time, suggestions and help. Additionally, I also thank my friends in eBiquity lab who were very enthusiastic about my work and have provided suggestions and constructive criticism. I will always be grateful to Bazookz for their constant support for the past two years. A special thanks for Alark Joshi for his help and motivation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Social Media are groups of the new online media where individuals are connected through communities and other open means of participation in sharing information and their participation in discussions. According to wikipedia (http://en.wikipedia.org/wiki/Social_media) "Social media describes the online technologies and practices that people use to share opinions, insights, experiences, and perspectives with each other". Presently Blogs, Social networks, wikis, podcasts, forums and content communities constitute to some of the important kinds of social media. Blogs are perhaps widely used by the internet users due to their ability to disseminate information and present their ideas on various topics. Webpages contains static information and does not have the feedback, which affects the user interaction. The ability to present blog content in the form of text along images, links to other webpages, audio, video and user comments distinguishes the blogs with other areas of social networks.

A blog , short for "web-log" , is a journal entry by individuals written in reverse chronological order. According to wikipedia (http://en.wikipedia.org/wiki/Blog) blogs provide information about a particular subject such as food, politics, sports, news and are also used more as online personal diaries of users, who generally called bloggers. The blog written by the bloggers are user generated contents which generally presents opinionated

views on a topic. Due to its low cost, user-friendly blogger software and availability to online users, blogging has become a powerful tool to express an opinion and critique on a subject. These blogs have increasingly become an important information source for the users ideas, sensitivity and sentiments. Thee subjective information in blogs helps in understanding a blogger's views and observations about various topics. Since bloggers are a good representative set of the entire population, its important to understand the sentiments, both positive and negative opinions about the topic to understand the public views in detail.

In this thesis, we address the problem of detecting sentiments from blogs. To address the problem we develpoed a modular domain independent framework for the extraction of subjective information from blogs. Currently, there is no sentiment analysis framework that presents a sentiment based analysis on the search query and presents the results according to the subjective sentences found in the them. Also, having a domain independent system would help in customizing the framework to the specific domain, thus projecting more information and more statistics about the specific domain. Our approach targets political blogs since politics involves representing and leading the people and conveying and convincing the community and obtaining and analyzing the opinions of the public. Sentiment analysis on this domain would be particularly helpful, given that blogging involves in the spread of opinions on a subject. A subjective sentences is for example *"I like Hillary for her hard work and perseverance"*, where as objective sentences may be *"Hillary is right handed"*. Also, Daniel (Drezner & Farrell" 2004) presents a detailed description on the power and influence of blogs on on American politics. With the elections around the corner, political blogs are vital to exerting and keeping political influence over society. We collect the urls from the related domain, parse the blog content from the new feed that we obtain when we pass the urls through the system for sentence level analysis. These sentences are passed through the pattern matching filters for matching the sentences according to the customized domain dependent patterns, nave bayesian filters based on the training

set data and the parts of speech filter for obtaining the sentiment oriented sentences which are later indexed in the date based multiple indexing approach. Also, the results that are obtained based on the query, which employs customized query based boosting are separated based on the sentiment orientation of the sentence. Hence, more commonly, we address the problem of developing a sentiment analysis framework based on the concept of modularity and thus keeping the domain independent approach for further expansion.

Our approach employs different filters for opinion extraction. Firstly, a custom developed domain specific pattern matching technique is employed. Since we based our domain on politics, we custom developed our patterns according to the political patterns that are found in blogs. We first tried the shallow approach based on bag of words was initially employed, though worked well had indexed everything that was in the bag of words including lots of objective sentences, thus not working as efficiently as desired. This approach gave us the intricate difficulties in recognizing the pattern matching techniques thus leading a way for developing a nave bayesian machine learning approach which trains the data based on the objective and subjective dataset that is also custom developed for politics. Also, a more general parts of speech approach is employed for trying to figure out sentences that adhere to the opinionated parts of speech approach obtaining sentiment oriented words. Additionally, we ventured into trend analysis for expanding the modular approach by viewing in a different perspective for analysis. For this approach we employed the ngram approach and the KullbackLeibler divergence (K-L divergence) methods for obtaining the hot and top trend topics that are significant from the obtained sentiment sentiments.

This work presents an robust and flexible framework for developing modular sentiment analysis toolsWe envision this system to working with various domains and if possible building a common framework that would culminate the different domains into one big domain, thus using one framework for all the different domains.

1.1 Background

Blogging by nature is about expressing and listening to opinions. Good sentiment detection tools, for blogs and other social media, specifically tailored to politics are an invaluable tool for today's politicians. The blogosphere has become a dominant force to express opinions and views and the politicians have expressed been very vocal about their views on a topic through the blogs. For example, Hillary Clinton recently expressed her views on the new bill *"The Paycheck Fairness Act"* through the power of blogs. Hence, the politicians have used blogs as a mode of expressing their opinions about a subject and also in understanding bloggers opinions and sentiments to analyze their performance. Another excellent example was the blogosphere showing their displeasure over John McCain's negotiations on immigration reform and they wanted him to apologize for that incident (http://www.nypost.com/seven/05222007/news/nationalnews/mccain_ sorry_ for_ outburst_ nationalnews_ charles_ hurt.htm) .

Currently, we have different search tools on the internet to give us the information about a topic, however, blog data is more about expressing views rather than giving information about it and hence a more sophisticated and a better way or organizing the data is required. For example, querying about *"George Bush"* in google would yield only the list of urls that contain information about him. However, if there is a system that would detect the sentiments in blogs and present them in user interactive format, then it would present a more clear picture to the politicians and public. So there is a need for a sentiment based framework that would provide a classification of opinions based on the sentiment of the blogs. Presently, there is no such framework that focuses on indexing and querying political blogs based on the sentiments of the blog. For example, suppose a user wants to find about a particular political issue, like the opinion on the arms act or information about the Iraq war, the normal blog based search engine would search the index for the topic and

output the results. However, it would be better if a sentiment based query search framework outputs the results to reflect the opinions of the blogger, thus giving a much deeper insight about the blogger community and its view a specific topic. We believe that such a framework would be useful by giving a different perspective on a topic to the user.

The reminder of the thesis is organized as follows. Chapter 2 covers the related work. Chapter 3 describes the details of our approach, heuristic and the important features of the framework. Chapter 4 elucidates on the experiments that were done and the conclusion and future work are presented in Chapter 5.

Chapter 2

RELATED WORK

This chapter surveys previous work that is related to our research contribution. A common theme in much of this work is building a framework for the political system and to develop a sentence level analysis on the blogs to index and query the system. For this reason, we enumerate and explain the related work based our area of research below.

2.1 Sentence level sentiment analysis

There has been a growing interest in detection and classification of sentiments upon Blogs, as shown by the 2006 NIST TREC Blog track (Craig Macdonald 2006).This is expected, given the size and rate of growth of the Blogosphere and justifies the need for a more intelligent use of the blog content. Hence a powerful and an organized system that would analyze the sentiments of the blog data at sentence levelwould be invaluable. Turney (Turney" 2002) proposed a simple unsupervised learning algorithm based on semantic orientation for classifying reviews on the web as "thumps" up and "thumbs down".The semantic orientation is calculates as the mutual information between the given phrase and the word "excellent" minus the mutual information between the given phrase and the word "poor".As shown in Pang (Pang, Lee, & Vaithyanathan 2002) different Machine learning techniques are analyzed and it is shown that unigram SVMs (Support Vector Machines)

do well for classifying movie reviews. However, blogs are not movie reviews and should not be treated like them as Engstrom (Engstrom 2004) had showed that the bag of-features approach is topic-dependent. A classifier trained on movie reviews is unlikely to perform as well on the reviews of automobiles or blogs. In addition to being informal, poorly structured, rife with spelling and grammatical errors, blogs are about potentially multiple topics from a wide variety of domains. Since topics can change mid post using a uni-gram SVM on complete blog posts to detect sentiment is inappropriate as it would pickup on sentiment words from other topics. Most importantly, users only want to see the relevant sections of a blog post in the search results from political sentiment retrieval engines, they don't want entire blog posts.

BlogVox opinion Retrieval system (Akshay Java & Mayfield 2007) retrieves opinionated blog posts based on the post level content by discriminating against spam blogs and incorporating SVM based system and integrating relevancy score to rank the results. Mullen and Collier (Mullen & Collier 2004) introduced an approach classifying opinionated sentences by incorporating several new information sources as features into Support Vector Machines. Wiebe, Wilson and Hoffmann's (Theresa Wilson 2005) approach to phrase level sentiment analysis involved in determining whether an expression is neutral or polar and based on this analysis, the sentence differentiates the polarity of the subjective expressions to positive or negative. Their approach involves identifying the contextual polarity of phrases based on the clue words and then disambiguating the collected phrases.

Wiebe and Riloff (Janyce Wiebe 2005) explores upon the idea of subjectivity analysis to improve the precision of information extraction system. Yu and Hatzivassilogou (Hong Yu) deals with differentiating opinions with facts in document level as well as sentence level. The approach involves a Bayesian classifier for discriminating between documents with opinions and describes three unsupervised statistical techniques for detecting opinions at the sentence level. Using minimum cuts (Pang & Lee 2004), Pang broke doc-

uments down into objective sentences and subjective sentences. The subjective sentences are then used as if they were the original document. Neither of these works use part of speech tagged n-grams, and Pang (Pang & Lee 2004) uses the unigram feature approach. Pang describes a novel idea of limiting the text characterization to the subjective portions of the document, thus employing the minimum cuts in graphs which facilitates cross-sentence contextual constraints. However, more future work is required for parameter selection techniques. However, the graph cut formulation yield better results for naive Bayes method and the use of SVM. Dave (Dave, Lawrence, & Pennock 2003) presented complete product reviews where trigrams outperform bigrams which in turn outperform uni-grams in the two tests for Naive Bayes Classifiers. However, rating inconsistency, sparse data and skewed distribution affect the performance of the system.

2.2 Filter analysis

Sentence level filters are becoming more prominent with sentiments because in a document level analysis, even if a sentence speaks positively or negatively about a topic, the sum of all the sentences would negate the effect of the single sentence, thus altering the polarity of the topic. Lawrence and Pennock (Dave, Lawrence, & Pennock 2003) introduces a opinion mining tool, that aggregates opinions about a given topic classifies them according to the subjective analysis. This is done by identifying the unique properties of the problem and developing a method for automatically distinguishing between positive and negative reviews. The classifier uses information retrieval technique for the extraction and scoring of opinionated sentences. However, the tool focuses mainly on movie reviews and not hence cannot be used as a domain independent tool. Additionally, mixed reviews introduce noise which deteriorates the performance of the system.

Eric and Cardie (Choi, Cardie, & Breck) focus on identifying the words and phrases

that express opinions in a text rather than the entire document. They employ conditional random fields and evaluate the approach at the expression level using a standard sentiment corpus. Das and Chen (Das & Chen) manual construction of discriminant word lexicons have been inspired the sentiment-based categorization of entire documents

We did not need special spam filtering technology, since we focused on spam free urls that was indexed in our database. Kolari's (Pranam Kolari & Joshi 2006)work on characterizing the splogs in blogosphere and employing the machine learning techniques for removing splogs (Pranam Kolari & Joshi 2003) have been efficient in removing splogs in blogosphere.

Work on Hearst (Hearst 1992) on classification of entire documents uses models inspired by cognitive linguistics. Hatzivassiloglou and Wiebe (Hatzivassiloglou & Wiebe 2000) investigated sentence subjectivity classification. They proposed a method to find adjectives that are indicative of positive or negative opinions based on the semantic orientation and gradability. Nasukawa (Yi *et al.* 2003) worked on the online sentiment classifier that employs natural learning processing techniques.

Chesley, Vincent and Srihari (Chesley *et al.* 2006) examine the novel idea of using linguistic feature, verb class information, and Wikipedia dictionary for subjectivity classification. Their classifications have improved the subjectivity classification accuracies when compared to the baseline established classification.

2.2.1 Trend Analysis

Trend analysis is another area that has received a lot of attention lately. Textmap (Izzet Zorlu 2005) analyzes various domestic and international documents to mine for entity references and analyze the juxtapositions between them. Yi and Niblack (Niblack & Yi) employs data mining techniques to detect sentiments by developing a webfountain system. We plan to extend and work on this area to develop it even further.

Chapter 3

PROPOSED APPROACH

This chapter deals with the proposed approach that's taken for the implementation of the framework. The core of this thesis will be the divided into eight parts which are used for experiment analysis.

1. Framework
2. Dataset
3. RSS Feed Collection analysis
4. Sentence analysis
5. Filters analysis
6. Additional Modules
7. Indexing
8. Searching

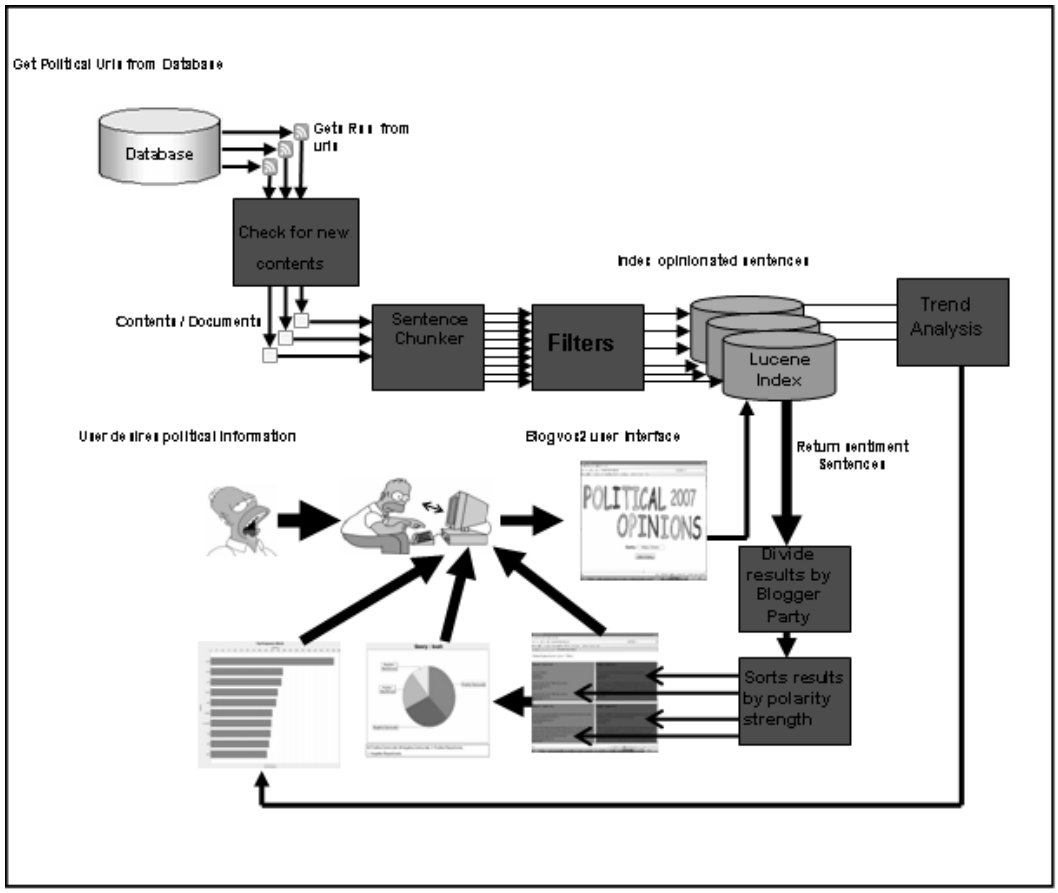

FIG. 3.1. The Architecture of Blogvox2 Framework

3.1 Framework

The overview of the system architecture is presented and we elucidate the basic working model of the architectural framework below.

The political urls that are collected are stored in our database. We have collected 3028 blogs for political analysis. For each url from the database, we collect the rss feeds, where the RSS is Really Simple Syndication. An RSS document(RSS), which is called a ”feed,” ”web feed,” or ”channel,” contains either a summary of content from an associated web site

or the full text. These feeds are collected based on the ROME parser and HTML parser(). ROME parsers are set of Atom/RSS Java utilities that fetches the data and the feeds of the urls for sentiment analysis where as HTML is a Java library used to parse webpages for content extraction.

The obtained rss feeds are sent through Lucene index(http://lucene.apache.org/java/docs) for obtaining the new blog contents and if the feed is new, it is indexed and the feed content is passed through the block for obtaining the blog content and for further analysis. However, if the feed is already indexed, then it is skipped as the rss feeds for each blog topic is unique and the blog contents would have already been indexed. After the rss feed passes through the checker, the contents from the feed is obtained and this blog content is stripped of the advertisement and other unrelated data based on the tidy jar and other unsupervised custom made cleaner and the obtained blog content is then sent to the sentence chunker for sentence level chunking of the obtained data. The sentence chunker is based on lingpipe (http://www.alias i.com/lingpipe) and a custom developed system based on punctuation which chunks the blog data into sentence format for further analysis. The obtained sentences are then passed through two filters namely the pattern recognizer filter and the nave bayes filter. These filters are mainly used for filtering out the objective sentences and classifying the subjective sentences into positive and negative sentences. These sentences are indexed in our custom made multiple indexer we create for each day. The metadata that is indexed are the date of blog, the blog permanent link along with its url, the contents of the blog and its polarity.

This index allows us to build different search applications on this framework. We developed a prototype two-panel positive and negative view and a prototype four-panel positive and negative view when we index the label of the blogs along with the rest of the metadata for showing the flexibility of the indexes. Also, graphical representation such as pie-chart for the two and four panel are presented for further analysis.

Additionally, we ventured into trend analysis where we computed the hot and top words based on the multiple indexes. Top words are words or topics that always a point of discussion in blogs. For example, in political domain, bloggers constantly discuss about ”George Bush”, ”Iraq war” and ”Arms act” and these are considered as the top topics. Hot topics are words or topics that are currently discussed which normally don't figure in top words as these are topics that are discussed normally due to the occurrence of some incident. For example, the recent massacre at ”Virginia tech” or the ”immigration bill” are some of the hot topics that figured in our results.

The framework we developed forms the base for our experimental analysis.

3.2 Dataset

We based our analysis on different dataset in politics. We also considered the twitter dataset (http://twitter.com) for analysis since ours is a modular and domain independent framework, we analyzed our system with different domains.

3.2.1 Lada A. Adamic Political Dataset

One of the dataset we considered is the custom developed political blog urls that contained 3028 of urls for analysis. We narrowed our urls for analysis to randomly selected 649 urls from which we obtained the rss feeds and other related data for further observations.

3.2.2 Lada A. Adamic Labeled Political Dataset

Lada A. Adamic provided a reference dataset of 1490 blog urls that was classified as democratic and republican for each urls. Some of their data were manually labeled, based on the incoming and outgoing post at the time of the 2004 presidential elections. We used this dataset for the development of the four panel sentiment view for projecting the

query results. Since all the other datasets did not have labels, these datasets proved that the framework is truly modular and we could attach a lot of other applications around the framework.

3.2.3 Twitter Dataset

This dataset was provided by twitter (http://twitter.com) and we used it in our framework since ours was a domain independent system, we wanted to index, query and search for results on an entirely different domain to realize the extent of its flexibility. This provided the much needed information on the ways to extend and develop the framework.

3.2.4 Spinn3r Dataset

Spinn3r (http://tailrank.com) provided live spam-resistant and high performance spider dataset to us. We tested our framework on this dataset since it was live feeds and we wanted to test our performance of sentiment analysis on these dataset for performance analysis and testing. We periodically pinged the online api for the current dataset of all the rss feeds. Although we had different domains that were provided to us, we chose the political domain for consistency with our other results.

3.3 RSS Feed Collection analysis

We fetched the rss feeds from the urls that was provided by our custom collected political urls and the Lada A. Adamic dataset. We employed Rome parser (http://wiki.java.net/bin/view/Javawsxml/Rome) for obtaining the rss feeds and the content of these feeds. These feeds are collected and indexed along with the other metadata like the date of publishing of the blog, the sentiment sentences that are obtained from the blog data content collected by rome parser and the parent url from which the permalink is

obtained. We used rome parser for the on our custom collected political urls, lada adamic dataset and the spinn3r dataset.

3.4 Sentence analysis

There has been a considerable work on sentence level analysis on opinion extraction. Sentiment detection in the general case is much more difficult than sentiment detection in one domain. For our domain independent sentiment detection framework, it is more appropriate to use sentence level opinion scoring instead of the original document level scoring modules since individual search results should be small and easy to display. Sentence level sentiment detection is different from document level sentiment detection because even though a document might not have any clearly opinionated sentences the sum of its parts could communicate a clear sentiment. In politics, this should not be much of a problem, since critics and supporters are often outspoken.

The content that is fetched is passed through the sentence chunker block. In this block, the content are broken down into sentences by using Lingpipe (http://www.alias i.com/lingpipe) and custom developed sentence chunker. Lingpipe is a java based linguistic analyzer which is widely used for information extraction and data mining. However, the sentence extraction has not been efficient as perceived as lingpipe training set data is trained on bio-medical literature corpus where the data is entirely different from blogs. Hence, effectiveness of sentence extraction was a factor. We added our own custom made punctuation based sentence extraction based on heuristic technique that is modified to suit the blog data on the top of lingpipe sentence extraction for better results. This is later passed through the sentiment filter analyzer for the extraction of opinionated sentences.

Choosing to use sentence level sentiment detection has implications for sarcasm detection. In document level sentiment analysis a sarcastic sentence would need to compete

with the lump sum of posts overtly expressed opinions in order to fool the classifier into misclassification. In sentence based sentiment detection sarcasm has no such obstacle to overcome and thus poses a greater threat. If a sarcastic sentence is misclassified it is more likely to be taken as genuine, misconstruing the writer's intent. Additionally, sarcastic sentences taken out of context and trained on, are much more likely to cause classification errors. One of the solutions was to use a more domain specific training data, which would help in churning out sentiment sentences. Although this is not a perfect solution the training dataset helps as in the case of politics and technology, the blogger is generally outspoken and blunt where as in the case of entertainment domain, since it caters to more general bloggers, comparatively, there would be more sarcastic sentences since the bloggers do not adhere to a specific style of writing. Sarcastic sentences are often wrongly placed as it is difficult to identify sarcasm. This has been another research area that needs focus and lots of work is going on in this field. (Julius Quiaot 2007)

We observed that sentences that have more than one named entities or topic are quite common in the domain of sports, politics and entertainment. For example, sentence like "I hate hillary, but I like Edwards", would confuse the training set and our pattern recognizer as in the example, the first part of the sentence speaks negatively about the first topic where as the second part of the sentence speaks positively about the second topic, but both the parts are in one complete sentence, which complicates the detection of sentiment analysis. One solution is to reduce the polarity score of the sentence by recognizing the number of named entity or topic words in a sentence. If there is more than one named entity, then the score of the sentence reduces since the analyzer is not able to attribute the polarity to any given named entity or topic.

3.5 Filters analysis

3.5.1 Filter Overview

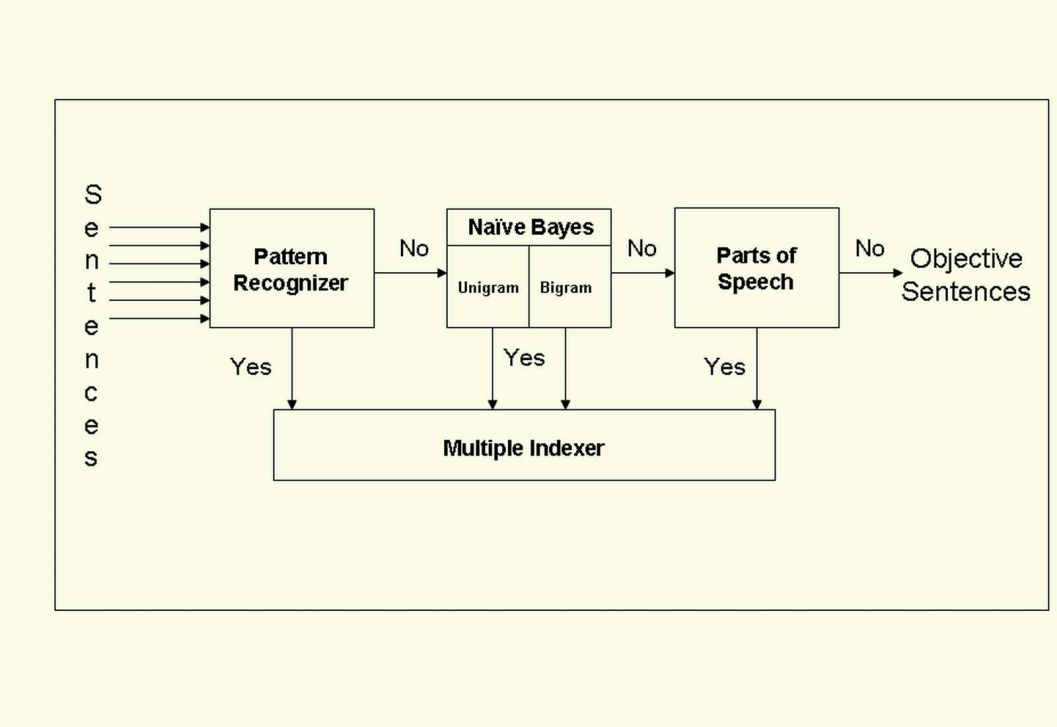

FIG. 3.2. Overview of filters for subjective sentence analysis

Our framework employs two different filters in addition to the three supporting modules for detection of sentiments in sentences. After extracting the sentence, we pass it through the sentiment filter for the analysis of opinions. When one of the filters classify it as subjective, then the sentence is indexed based on its polarity. If the sentence passes through all the filters without being classified, then we mark it as objective sentence and discard the sentence.

Firstly, we pass the sentence through the pattern recognizer filter. Pattern recognizer

checks the sentence for any sentences that follow the subjective patterns that are custom provided and if they follow a pattern, then the sentence is indexed based on the polarity, else the sentence is passed through next block. However, if the sentence is not indexed, then it is passed through the nave Bayes block for further analysis. Here, the nave bayes method is selected for sentiment extraction of sentences. This block depends on training set data and based on the machine classification which classifies the sentence as subjective or objective. If the sentence isn't indexed in the subjective index, then it passes to the next additional kind of filter called parts of speech filter. The sentences are tagged with the parts of speech tagging. The ngrams (unigram and bigram) are also parts of speech tagged and the sentence is passed through this block. If the sentence is found to be subjective, then the sentence is indexed, else the sentence is skipped and a new sentence undergoes the entire procedure, till the end of the blog is reached.

Identifying named entity (or topic filter) and "bag of words" blocks support the other filters by increasing the information of the indexed sentence. Just before the sentence is being indexed (after the filters classify them as subjective), the sentence passes through the named entity filter for the identification of more than one named entity present and then through the bag of words filter for the identification of the strength of the sentence (how polar they are). While the former reduces the score if there is more than one named entity (or topic), the latter provides a metric for subjective sentences.

3.5.2 Pattern Recognition Filter

Pattern recognizer filter is custom developed domain based filter for identification of patterns. For example, a common sentence in political domain contains subjective sentences such as *"I am proud that I voted for Hillary"*. Sentences like that are very common in the case of political domain where as sentences *"I like Hillary"* are common in every domain. Pattern recognizer is required to be domain specific, and since we had chosen

politics as our domain, based on the heuristic method, we identified 95 positive and 162 negative patterns for our framework.

3.5.3 Nave Bayes Filter

Nave Bayes classifier is a simple probabilistic classifier based on applying the Bayes theorem with strong (nave) independent assumptions (Mccallum & Nigam 2001). This approach to text classification requires to assign a document *"d"* which are labeled sentences that need to be trained and classified according to the polarity. Pervious work (Riloff, Wiebe, & Wilson 2003)(Pang, Lee, & Vaithyanathan 2002) (Hong Yu) performed well under nave bayes classifier for subjective recognition so we used nave bayes as our learning algorithm for sentiment recognition.

Training set involved in manually classifying subjective and objective sentences and reclassifying subjective sentences into positive and negative sentences. We trained the classifier using the initial training set that was available online (the movie review dataset) and our custom developed political dataset. There are several features like the strong positive negative and neutral sentences in political domain which is classified manually and the polar and neutral sentences that are trained from the movie database (Bo Pang & Vaithyanathan 2005). Our custom developed political dataset contained 273 negative sentence, 320 objective sentences and 178 positive sentences. We enhanced the training set data with the movie dataset which contained 5331 positive and 5331 negative classified sentences.

For a document *"d"* which consist of positive, negative and neutral sentences, the probability of obtaining the positive, negative and neutral sentences are as follows. For a sentence, we chunk it into individual words and for each word we check the probability of the occurrence in the ngram analysis. Having obtained the probability of the words, the sum of all the probability in that sentence is obtained and averaged. Here, these averages

that are obtained are compared with a threshold of ".6" and if the average is more than the threshold, we index the respective sentence as subjective.

Assume document *"d"* is a collection words in a sentence W1, W2, W3 ... Wn. Hence,

$$\text{d} = \sum_{i=0}^{n} Wi \text{ (where n = number of words)}$$

for each i, the value of Wi = p(i), where p(i) is the probability of the word *"i"* obtained from ngram analysis. Hence, the document is subjective

$$\text{if } d^p = \sum_{i=0}^{n} Wi, \text{ where Wi = p (i) and d} > .6$$

Hence, this is a modified version of naive Bayes classifier since, in naive Bayes method, the probability is multiplied, but we take an average value of the probability of words. This is because, since we base our analysis on positive, negative and objective sentences, if a sentence is positive, and if it contains one word thats is negatively inclined, then the value of the probability reduces drastically and thus the sentence will not be classified as subjective sentence. Hence after observing the system, we averaged the probability of words for better results.

3.6 Additional Modules

Two modules are added along with the existing filter modules to refine the filtering of subjective sentences. Part of speech tagging is employed with the n-grams, where as the named entities are used to parse the subjective sentences for filtering out sentences that do not clear if the subjective sentence has more than one named entities.

3.6.1 Part of Speech tagging

A part of speech is an addition to the existing model that is added for churning out opinionated sentences. Part of speech features have been effectively used in opinion classification (Theresa Wilson 2005). In a document, subjective sentences are more probable to

occur when there are more adjectives and adverbs as compared to sentences without these, since adjective and adverbs denotes behavior of entities and events. According to (Pang, Lee, & Vaithyanathan 2002) tagging adjectives alone do not give enough information and would perform badly as shown if tagging is done and used for analysis. However, tagging unigram with parts of speech gave better results as compared to the tagging of adjectives alone. We used the unigram and bigram techniques for our analysis. For recognizing the parts of speech and tagging it, we employ lingpipe parts of speech tagging (http://www.alias i.com/lingpipe) where the tokens are sequentially labeled with syntactic labels.

3.6.2 Bag of Words Filter

Bag of words filter is the final filter block that is developed by compiling a positive and a negative list of all the words that frequently occur in a subjective sentence. We identified 2712 frequently occurring negative bag of words list and 915 positive word list which is used for our analysis. For example, if a sentence like *"Hillary Clinton is not only good but also a well-organized women"* is considered to be more positive than *"Clinton is a good women"*. Hence the former gets more score in the metric system as compared to the latter sentence.

Having bag of words in filter will index lots of objective sentences as shown in the experiments. An example such as *"I like Hillary Clinton"* is a positive sentence due to the term *" like "*. However, sentences *"She looks like Hillary Clinton"* do not necessarily convey if the subject is subjective, however, the bag of words would index this sentence. Hence we do not use the bag of words as a filter module, rather we use it for the computation of the strength of the subjective sentence.

3.6.3 Named Entities

Named Entity filter is an addition filter that doesn't classify subjective sentences, but the identified sentences are passed through before being indexed. Sentences like *"Obama is inefficient as compared to Edwards who is hardworking and punctual"* has two named entities within a sentence, one speaking good where as the other part speaks bad about the person. These are difficult sentences that need a more careful attention, and one of the solutions is to push the sentence score down since the framework is unsure about its results.

3.7 Indexing

Indexing is one of the more important components for information retrieval. We introduce a novel technique of multi-indexing based on the date. We maintain different index for different days to maintain modularity and to reduce complexity and the indexing is done based on Lucene (http://lucene.apache.org/java/docs).

Using sentence based sentiment analysis allows us to index and store opinionated sentences in Lucene for later retrieval. Therefore, opinion detection is not time critical but must be done without knowing what the topic of the opinion is. Since Lucene returns results based upon the presence of the query word it is entirely possible that an index sentence could be opinionated about a topic other than the one Lucene retrieved it for. How often this occurs is query and time dependent. If a politician's name is queried and they are in the middle of a campaign against another politician it is more likely that an error will occur since competing candidates are often mentioned in the same sentence.

The index maintains the subjective sentence, its url and permalink, the date and the polarity of the sentence. We also maintain the url- rss indexing according to the date where the new feeds that are analyzed are indexed in the urlindexer for avoiding re-analyzing the data. In this case, the data is being analyzed and data and other information is indexed in

the multiple indexed that is being developed, where as the url-rss indexing maintains the permalinks in a separate index. Indexing is used instead of a database to avoid delays in fetching the data.

3.8 Searching

Searching is another important component of our framework. Having obtained the index framework, we can extend the framework by developing different applications by extending the search framework. The basic framework for the search contains the query and the result-set based on the query. The query is boosted, for example a query like *"Apple I-Phone"* would search a query that has "apple" and "iphone" in a sentence together. The search involves a query that contains both the words in a sentence within a gap of 10 words. Since this is a sentence, a gap of 10 words is a heuristic that is considered as the sentence boundary would contain the words within that range. Also, individual words that contain "apple" and "iphone" is searched and the results are projected, however these results have a score less than the query that contains both the words.

We developed a prototype two panel and a four panel view of the results. The two panel view divides the results into positive and negative view according to the opinions of the sentence, where as the four panel divides the results according to the opinions and the labels that's being indexed along with the results in the multiple indexer.

3.9 Trend Analysis

We tried to analyze the top words within the opinionated system. Here, we tried to extract the top topics and the hot topics from the multiple-indexer which is indexed with opinionated sentences. This analysis is to comprehend the results about the important topics are and what the most talked about topics are in the blogosphere. We present two different

kinds of analysis of the topics, the top topics and the hot topics for study which is based on the opinionated indexed information provided by the framework.

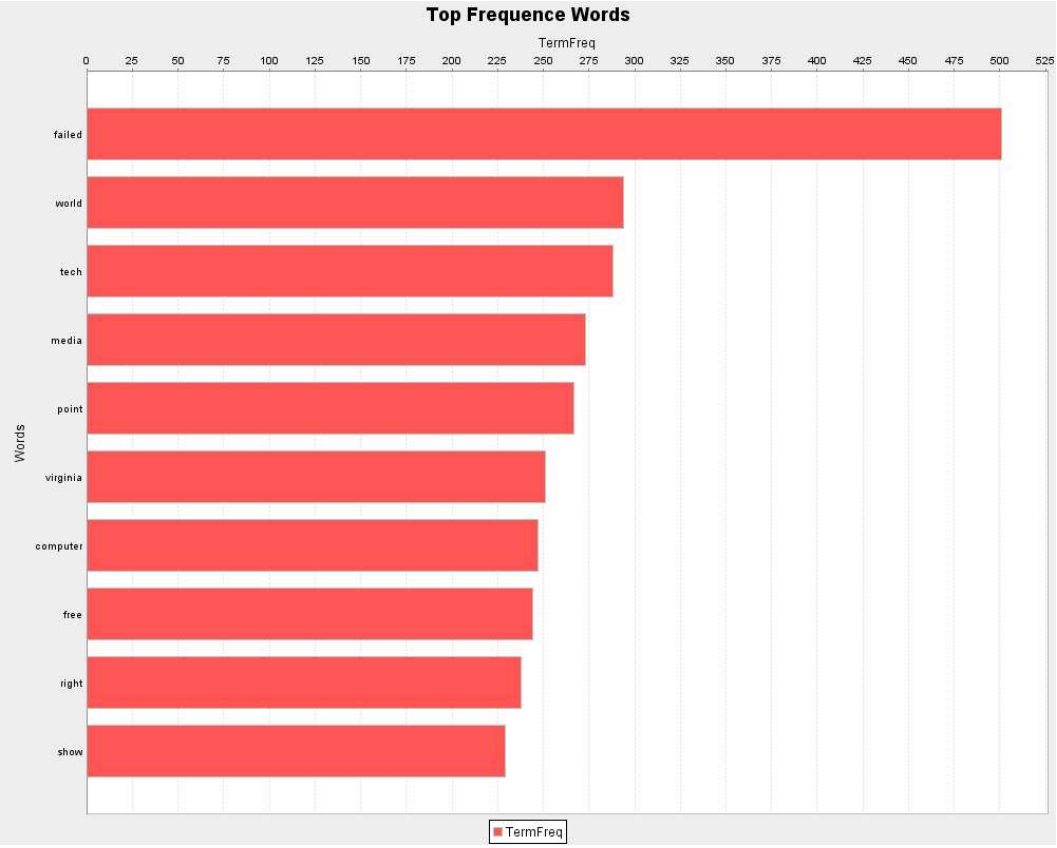

FIG. 3.3. Top Topics results from our Framework

Top topics are topics that have always been in the point of discussion in the blogosphere. Since we concentrate on the political domain, an excellent example is the president of USA, *"George Bush"* who is always talked in the blogosphere. Another example as shown in the figure **??** thats obtained from our frame work which was emph"Virginia tech", during the shootout at the campus, where for the bloggers wrote about them the incident for over a week. Hot topics are topics that are currently in the point of discussion with respect

to the blogosphere. Examples for hot topic include the death of *"Boris Yeltsin"* and the recent news about the *"immigration bill"*.

3.9.1 Top Topics

We compute top topics from the multiple-index that was indexed by the framework. This is done by employing Lucene (http://lucene.apache.org/java/docs) frequency search which projects the top terms in the indexer. We employed heuristics to remove the common occurring words in English reference (http://www.cs.cmu.edu/ cburch/words/top.html), dates and date related words, numbers since these words show bias towards the blogosphere. We based our heuristic to screen out words or topics that are less than three words. We present a live graph of the results as a part of the framework application, to present a clear picture on the topics that are discussed.

3.9.2 Hot Topics

We compute the hot topics by employing Kullback-Leibler divergence, also known as KL divergence (http://en.wikipedia.org/wiki/Kullback Leibler_ divergence)along with the current index, in addition to the pervious indices. K-L divergence is a measure of two probability distribution, the true probability distribution theory and the arbitrary distribution theory (http://en.wikipedia.org/wiki/Kullback Leibler_ divergence). True probability distribution theory is the numerical value of the current frequency of occurrence of the topic where as the arbitrary distribution is the value that is computed averaging all the previous occurrences of that topic.

For computing the hot words, we computed the average value of all the words according to their occurrences in the multiple indices. After obtaining the values, for the current date, we compute the KL divergence according to the formula given below.

Let the KL divergence be "D_{kl}" and the true probability distribution be "P" and arbitrary value be "Q". The KL divergence is

$$D_{kl}\ (P \parallel Q) = \sum_i P(i) log(P(i)/Q(i)).$$

We calculated the divergence according to this formula and the hot topics are words that have a high divergence ratio. For example, "Virginia tech" and "Immigration" received a high divergence ration when they made headlines. This is because both the topics weren't discussed much by the bloggers, but when they made headlines, there was a high divergence ratio due to their sudden discussion in the political domain.

Chapter 4

EXPERIMENTS

This chapter deals with the details of our experiments that were conducted during the development of the framework to demonstrate the feasibility and effectiveness. We describe the motivation behind focusing our framework on political domain, different parameter consideration and the experiment computation and results.

4.1 Domain choice parameters

Political domain was our primary focus, we also considered the twitter domain since our system is a domain independent framework. One of the goals of this framework is to develop the system in such a way that the index of the opinionated sentences are used to develop different applications, like the two and the four panel view, a domain like politics is a perfect choice as one of the dataset was labeled, which was needed for the four panel development of the system.

The following are some of the reasons for selection of the domain.

1. Currently, there is no framework for the political domain. Since blogs are a way of putting forth a view and commenting about that view, a framework in this domain would help the politicians and general people for a more interactive communication.

2. After some manual analysis of political blogs, we observed that politicians and peo-

ple writing about politics are more straightforward compared to other domains. This helped our analysis since, for example, a negative review was not as sarcastic as other domains. Since, there are more challenges and work going on in dealing with sarcastic comments, political domain had less of these, which helped us to concentrate on other sentiment analysis problems.

3. Lada A. Adamic dataset contained the labeled set for political domain, which helped us to create the four panel view. This was particularly efficient for analyzing the opinions in a labeled set, which presented more information about the query.

4. From the business standpoint, having a framework on politics would be extremely beneficial since it would present a clear insight on the public views about a person or topic. This is important particularly because, bloggers constitute a significant number of the population's view and trend.

We also ran the framework with the twitter dataset that was available to view the system performance. Twitters are microblogs which limits the data up to 140 words. This meant we had a more clear and mostly spam-free dataset for analysis.

4.2 Experiment parameter

4.2.1 Datasets

As mentioned in chapter 3, we considered four different datasets. However, we consolidated our data to a more specific dataset, the Lada A. Adamic dataset (Lada A. Adamic 2005) for experimental analysis. This dataset contained 3028 blog urls related to blogs with varying feeds. Since analysis of these many blogs is not feasible as it required human intervention,we restrained the dataset to a lesser number for experimental analysis and to reduce noise that's occurs due to the spam blogs.

4.2.2 RSS Checker

We also only considered the new feeds and didn't consider the updated feeds that might occurred. Once we indexed the rss in the indexer, we neglected the updated rss feeds, since they required us to update our multiple indexer which delayed the framework.

4.2.3 Sentence Chunker

There can be various heuristics that can be considered during the chunking of the sentence. We chose to include our custom based chunker on the top of lingpipe sentence chunker, where ours captured sentences that contained full stop, question mark and exclamation mark as the sentence completion.

4.2.4 Filters

Our framework passes each sentence one by one through each filter, where even if one filter indexes the sentence, we continue with another sentence. Later, as a part of future work, we plan to combine the filters so that the sentences will undergo a more finer classification of sentiments and then indexes. Since our current framework indexes sentences that are opinionated, a compromise on the finer classification is done to improve recall and in this way we negotiate on precision, which we plan to improve by combining the filters.

In the nave bayes method, we do not exactly follow the mathematical equations that are given in the method. Rather, we add all the probabilities of the words in the sentence, instead of multiplying them, since according to analysis, the value of the sentence goes down if the sentiment word is slightly different as compared to the entire sentence, where by the value of the sentence reduces drastically. A clearly example is *"Obama is a great guy with only some faults"* is a positive sentence. However, the occurrence of faults will deteriorate the value of the sentence when multiplied rather than adding the probability of

words.

4.2.5 Trend analysis

Top terms are obtained by removing the words that bias the results, like the HTML tags, blogs related to the dates. Also, we eliminated the common words in English (http://www.cs.cmu.edu/ cburch/words/top.html). An heuristic of considering topics more than three words are only considered.

4.3 Experiment Results

We present the results of our experiments in this section. The following section considers different parameters and their effect on the accuracy of classification. We try to comprehend the reasons behind the results and the methods to try and improve the experimental results.For the experiments below, the total sentences are sentences that are considered by the framework after elimination of the HTML tagging and using stop words. We manually read the blogs and picked out the subjective sentences from the blog content. This formed the basis of our experiments

4.3.1 Pattern Matching Analysis

Our customized Pattern matching technique is employed for indexing the subjective sentences to analyze the performance. The results in figure 4.1 indicate that pattern matching techniques requires more analysis and improvement in order to capture subjective sentences. For this evaluation, we added only the pattern matching filter and removed other filters for analysis. As shown, the framework captures subjective sentences that adhere to the specific pattern that is in the custom compiled list.

The problem with this approach of including only the pattern matching technique

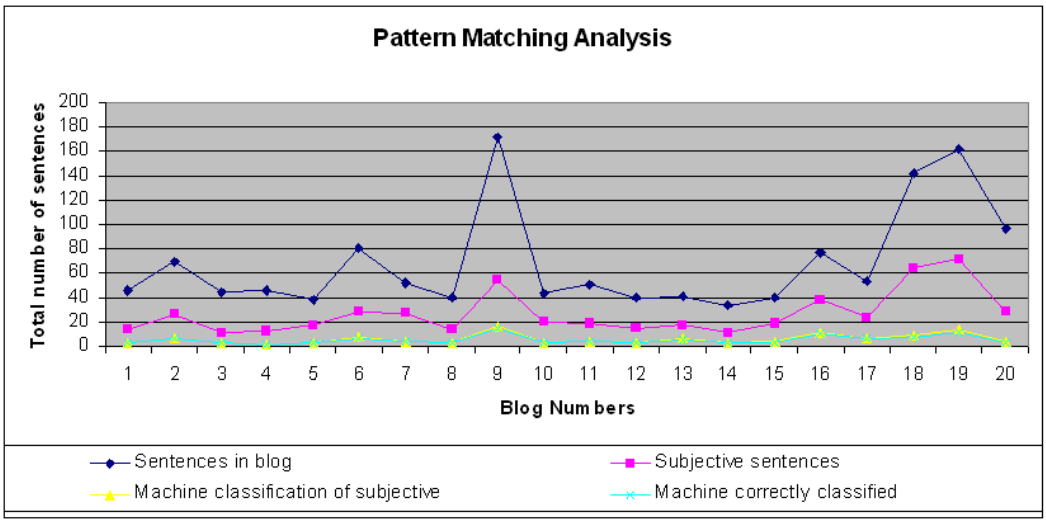

FIG. 4.1. Graph representing the analysis on Pattern Matching techniques, which do not show favorable results

is that patterns that follow that custom compiled list of positive and negative patterns are indexed. For example, in one of the political blog *"I like Hillary Clinton"* is a sentence that matches our custom compiled positive pattern *"I like *"*. However, as noticed, a sentence that the framework encountered *"I always liked Bush"* didn't get indexed since there is no pattern that matches from our custom pattern matching list. Hence a better approach would have been the use of develop more domain specific patterns and include various combinations of the compiled list. Also the use of stemming should improve the results.

Evaluation Metrics We now discuss the various evaluation measures computed from Figure 4.2 and using definitions from confusion matrix (http://en.wikipedia.org/wiki/Confusion_matrix). We use positive to denote subjective sentences and negative to denote objective sentences in order to compute the standard terminology.

$$Accuracy = 58\%$$

		Predicted	
		Objective	Subjective
Actual	Objective	98.44	1.55
	Subjective	82.04	17.96

FIG. 4.2. Confusion Matrix for Pattern Matching

Recall (True Positive Rate) = 18%

False Positive Rate = 2 %

True Negative Rate = 98%

False Negative Rate = 82%

Precision = 92%

This shows that the though precision rate of 92% is very high, the accuracy of 58% and recall of 18% is very less as compared to the other filters.which is why we incorporate Nave Bayes method, a machine learning technique to improve our results.

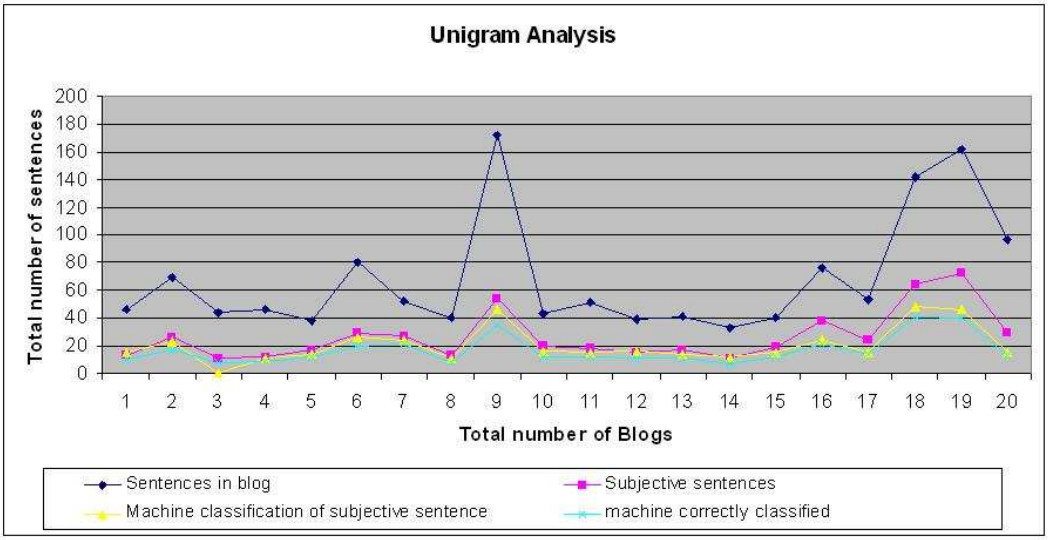

FIG. 4.3. Graph representing the analysis on Unigram techniques, which produces better results

4.3.2 Nave Bayes approach - Unigram Analysis

We base our analysis on unigram filtering analysis. Here, the unigram technique is employed for indexing the subjective sentences to analyze the performance of the framework. The results in figure 4.3 indicate that unigram analysis techniques is an better addition and improvement over the custom developed pattern matching techniques to capture subjective sentences. For this evaluation, we added only the pattern matching filter along with the unigram for analysis of the subjective sentences. As shown in the graph, the framework captures subjective sentences that adhere to the specific pattern and the sentences that contain high probability ratio which indicates the presence of subjective sentence.

However, more analysis on the training data is required in order to further improve the system. A better technique would be to custom compile the training set data according to the domain, which will improve the results. For example, we had compiled our list of

training set data that was customized to political dataset. However, we also included the movie dataset (Lada A. Adamic 2005) along with the training set. Though this helped the recall and precision, a training set data which focuses on political domain alone could improve the framework even further. Hence, we extend the unigram to bigram for a more detailed analysis.

		Predicted	
		Objective	Subjective
Actual	Objective	90	10
	Subjective	37	63

FIG. 4.4. Confusion Matrix for Unigram technique

Evaluation Metrics We discuss the various evaluation measures computed from Figure 4.4 and using definitions from (http://en.wikipedia.org/wiki/Confusion_ matrix). We use positive to denote subjective sentences and negative to denote objective sentences in order to compute the standard terminology for unigram analysis.

Accuracy = 77%

Recall (True Positive Rate) = 63%

False Positive Rate = 10%

True Negative Rate = 90%

False Negative Rate = 37%

Precision = 86%

This shows that the accuracy of 85%, recall of 80% and a precision of 63% is much better as compared to the earlier pattern matching filter and are comparable to the results

that obtained in (Pang, Lee, & Vaithyanathan 2002). Precision could be improved, as already mentioned, on more focusing on obtaining domain specific training dataset.

4.3.3 Nave Bayes approach - Bigram Analysis

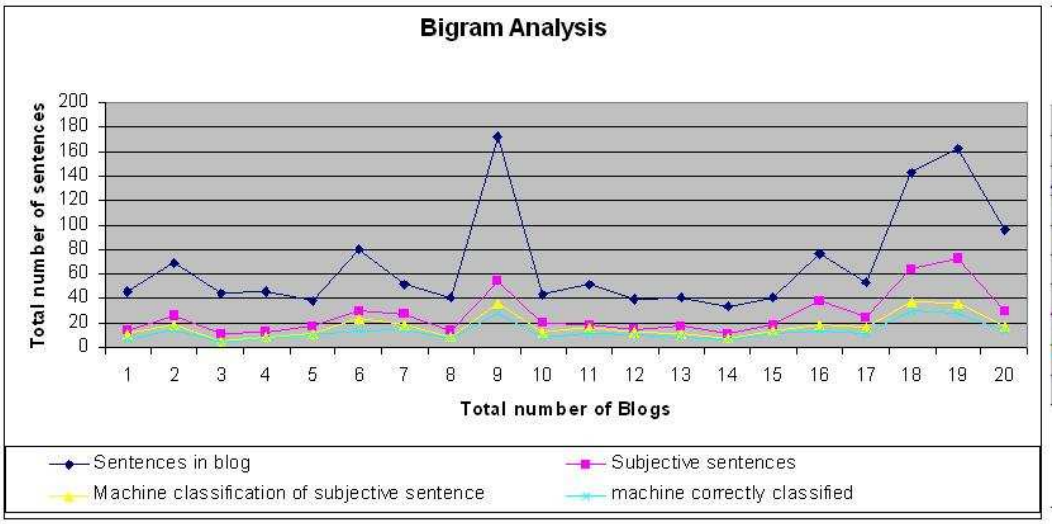

FIG. 4.5. Graph representing the analysis on bigram techniques, which produces better results, but not as better than unigram technique

We focus our analysis on bigram filtering analysis. Here, the bigram technique is employed for indexing the subjective sentences to analyze the performance of the framework. The results in figure 4.5 indicate that bigram analysis techniques also yield better improvement over the custom developed pattern matching techniques, but not as much as unigram analysis to capture subjective sentences. For this evaluation, we added only the pattern matching filter along with the bigram for analysis of the subjective sentences. As shown in the graph, the framework captures subjective sentences that adhere to the specific pattern and the sentences that contain high probable ratio which indicates the presence of subjective sentence.

However, as mentioned earlier for unigram analysis, more training data is required in order to further improve the system. A better technique would be to custom compile the training set data according to the domain, which will improve the results. We custom compiled training set data on political blogs, and for more data, we included the movie dataset along with the training set. Though this helped the recall and precision, a training set data which focuses on political domain alone could improve the framework even further. Hence, we extend the unigram to bigram for a more detailed analysis.

		Predicted	
		Objective	Subjective
Actual	Objective	90.5	9.5
	Subjective	51.6	48.4

FIG. 4.6. Confusion Matrix for Bigram technique

Evaluation Metrics We discuss the various evaluation measures computed from figure 4.6 and using definitions from (Pang, Lee, & Vaithyanathan 2002). We use positive to denote subjective sentences and negative to denote objective sentences in order to compute the standard terminology for bigram analysis.

Accuracy = 69.25%

Recall (True Positive Rate) = 48.5%

False Positive Rate = 9.5%

True Negative Rate = 90.5%

False Negative Rate = 51.6%

Precision = 83.6%

This shows that the accuracy of 73%, recall of 77% and a precision of 47% are comparable to the results that obtained in (Pang, Lee, & Vaithyanathan 2002),although precision deteriorates. Also, compared to unigram, bigrams do not perform as well as unigrams. More focus on obtaining domain specific training dataset is required. We now analyze appending both the unigram and bigram analysis for evaluations.

4.3.4 Nave Bayes approach - Unigram + Bigram Analysis

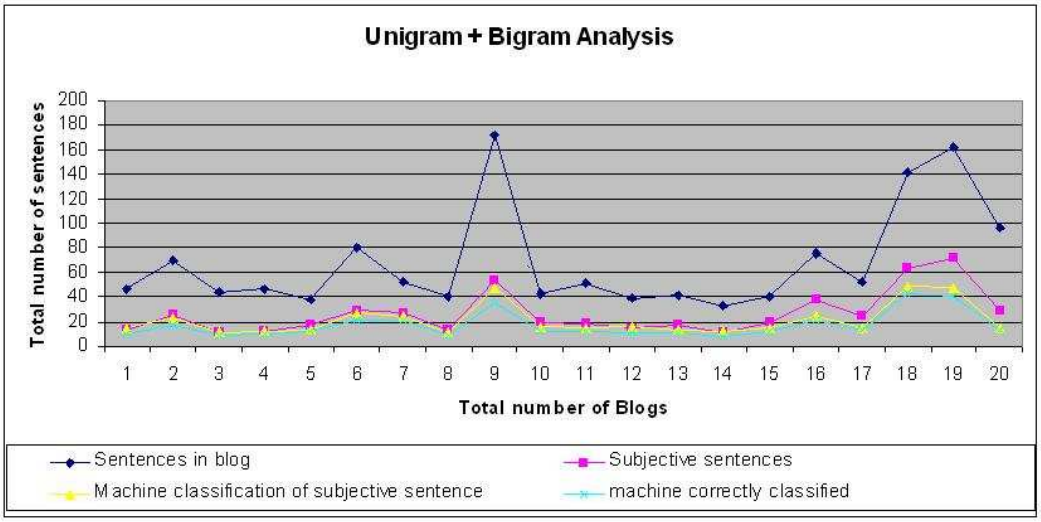

FIG. 4.7. Graph representing the analysis on combining unigram and bigram techniques

We focus our analysis on combining unigram and bigram filtering analysis. Here, unigram and bigram analysis techniques are employed for indexing the subjective sentences to examine the performance of the framework. The results in figure 4.7indicate that bigram analysis techniques also yield better improvement over the custom developed pattern matching techniques, but not as much as unigram analysis to capture subjective sentences. For this evaluation, we added the pattern matching filter along with the unigram and bigram

techniques for analysis of the subjective sentences. As shown in the graph, the framework captures subjective sentences that adhere to the specific pattern and the sentences that contain high probability ratio which indicates the presence of subjective sentence.

Performance has not improved much as compared to the unigram analysis, since; most of the sentences that are subjective are already indexed by the unigram techniques. Hence bigram only slightly changes the performance of the framework.

		Predicted	
		Objective	Subjective
Actual	Objective	90.05	9.95
	Subjective	35.92	64.08

FIG. 4.8. Confusion Matrix for Unigram and Bigram combined technique

Evaluation Metrics We discuss the various evaluation measures computed from Figure 4.8. We use positive to denote subjective sentences and negative to denote objective sentences in order to compute the standard terminology for bigram analysis.

Accuracy = 77%

Recall (True Positive Rate) = 64%

False Positive Rate = 9.95%

True Negative Rate = 90.05%

False Negative Rate = 35.92%

Precision = 86.55%

This shows that the accuracy of 79%, recall of 80% and a precision of 63% are comparable to the results that obtained in (Pang, Lee, & Vaithyanathan 2002) , although precision

still deteriorates. However, others are similar to the unigram and bigram techniques.

4.3.5 Parts of Speech Analysis

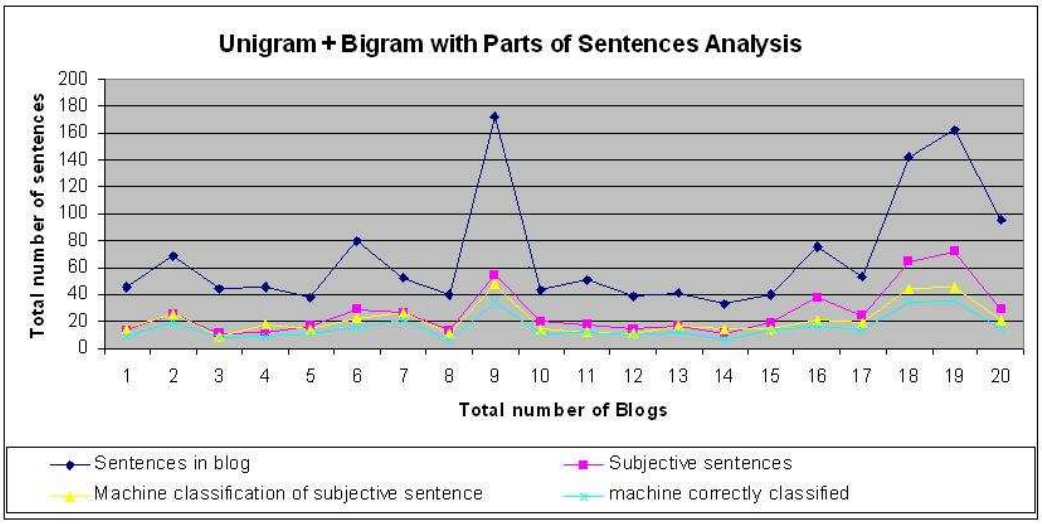

FIG. 4.9. Graph representing the analysis on introducing Parts of Speech techniques

Here, we analyzed the unigram and bigram techniques with Parts of speech tagging. For this evaluation, we added the parts of speech tagging matching filter along with the unigram and bigram techniques and removed all the filters that were not tagged to analyze the performance. The results in figure 4.9 indicate that this technique did not improve the performance as expected. As shown in the graph, the framework captures subjective sentences that adhere to the specific pattern and the sentences that contain high probable ratio which indicates the presence of subjective sentence.

Performance has not improved much as compared to the non tagging techniques. On examination, the tagging narrowed down the ngrams to specific tagging results, which affected the performance. For example, *"idiotic$JJ"* represents that idiotic is an adjective.

Hence in a sentence, words with adjective tagging are alone indexed as polar sentences and other sentences are screened out. Hence, this tagging technique doesn't perform as expected. However, an improvement could be done if this tagging is employed along with the other non-tagged filters for analysis

		Predicted	
		Objective	Subjective
Actual	Objective	86.7	13.3
	Subjective	40.45	59.54

FIG. 4.10. Confusion Matrix for Parts of Speech Tagging technique

Evaluation Metrics We discuss the various evaluation measures computed from Figure 4.10. We use positive to denote subjective sentences and negative to denote objective sentences in order to compute the standard terminology for bigram analysis.

Accuracy = 73%

Recall (True Positive Rate) = 60%

False Positive Rate = 13%

True Negative Rate = 87%

False Negative Rate = 40%

Precision = 81.7%

This shows that the accuracy of 82%, recall of 74% and a precision of 58% which is not as efficient as the other non-tagging techniques

Table 4.1. A Complete overview of the analysis

Filter	Accuracy	Recall	False Pos	True Neg	False Neg	Precision
Pattern matching	58%	18%	2%	98%	82%	92%
Unigram	***77%***	***63%***	***10%***	***90%***	***37%***	***86.65%***
Bigram	69.25%	48.5%	9.5%	90.5%	51.6%	83.6%
Unigram + Bigram	***77%***	***64%***	***9.95%***	***90.05%***	***35.92%***	***86.55%***
Parts of Speech	73%	60%	13%	87%	40%	81.7%

4.3.6 Experimental results from sentiment analysis

From the above table 4.1, we conclude that ***unigram*** perform better than most of the other experimental analysis. We thus propose that the unigram filter along with the domain specific pattern recognizer would outperform other machine learning techniques.

4.3.7 Threshold for Nave Bayes Analysis

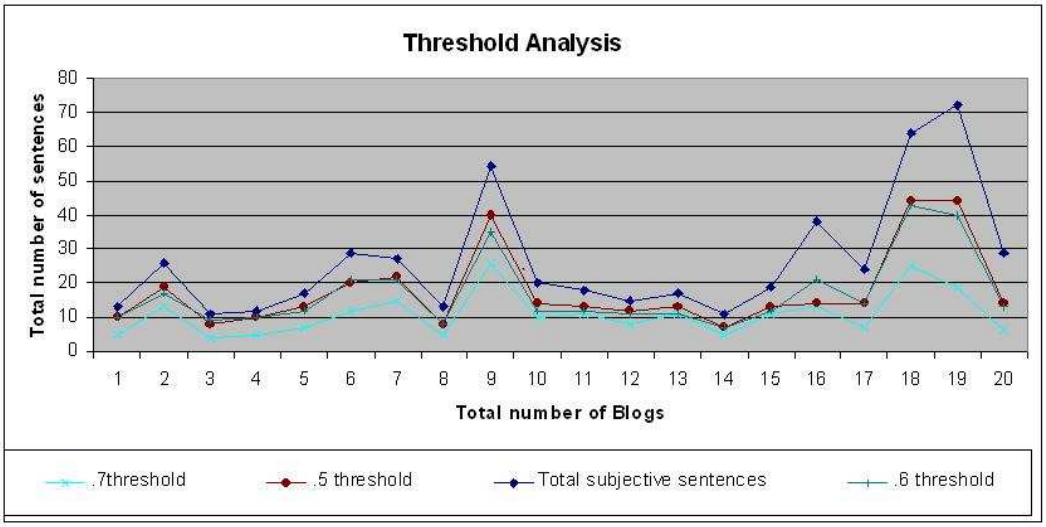

FIG. 4.11. Threshold evaluation for Nave Bayes technique

We based a numerical value of *".6"* for screening out sentiment sentences in our nave

bayes method. For example, in a sentence, each word has different probabilities based on the nave bayes analysis. Hence a average value of the sentence is considered where in a value of more than .6 is considered to be subjective sentence. We arrived at this value by analyzing different threshold values as shown in the graph 4.11. The optimum solution was .6, which yielded better results as compared to the .5 and .7 threshold values. Though the graph shows .5 to be better, on examination we found that it also indexes lot of objective sentences which will deteriorate the performance of the framework. Hence we chose a threshold value of .6 for our analysis.

Chapter 5

ONTOLOGY REPRESENTATION

This chapter deals with the motivation and approach of representing subjective sentences in ontology format.

5.0.8 Motivation

Our motivation to represent the results obtained from Blogvox2 stems from the fact that we required subjective and objective political dataset for training nave bayes approach of classification opinionated data. Also, many popular machine-learning algorithms require labeled training data to function. We initially obtained these dataset from (Pang, Lee, & Vaithyanathan 2002). However, this was a dataset that was customized to movie review which didn't present a clear picture based on the political domain. Also, we couldn't find any sources that had a subjective and objective classification on political domain. So we manually classified subjective and objective sentences that were customized to political blogs. This creation of these sets of labeled training data requires hand labeling a statistically significant number of instances. This process is time consuming, tedious, and separate from the problem that that machine learning is being used to solve. It is difficult for a researcher to find appropriate labeled training data in their domain, either no such datasets are publicly available, or they aren't in a format that is both recognizable and usable.

Also, problem results from the difficulty of labeling the data, and the absence of a flexible standard format in which to publish said data. Existing formats are too weak to provide the machine learning community with the features they need. Common formats include ”arff” and ”csv” both of which rely on commas to separate different features of an instance from each other. First, by breaking down an instance into features and leaving out the raw data a researcher has mandated the feature set that subsequent researchers must use even if they are inappropriate for the second researchers task. Second, by using commas to separate an instance's features these formats require the use of escape character sequences to represent commas. Third, these formats do not describe the semantic relationship between their features.

This meant that there is a requirement of obtaining all these information on web which would help other researchers to use the information for further analysis. For this reason, we developed an ontology file for sentiment classification which can be extended to other domains that would be available on web for further studies.

5.0.9 Approach

These problems can be solved using semantic web ontologies for labeled training data. Since no such ontology exists yet I propose the following top level ontology. This ontology for labeled training data allows the creator of the dataset to distinguish between raw data, features, and class labels by making them each distinct classes. The comma separated values format might allow a user to label the different fields with class names that imply which column is a feature and which is a class label, but it provides no way to show for certain. Additionally, the Feature class and the Class Label class can be subclassed using inheritance. Furthermore, the cardinality restrictions OWL is capable of fit our needs perfectly.

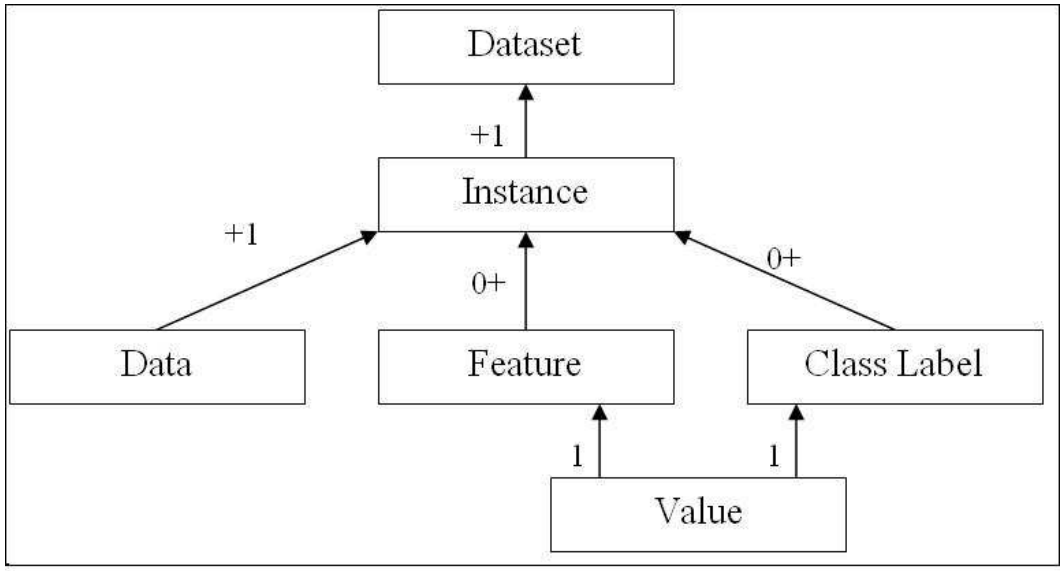

FIG. 5.1. Ontology Representation of Subjective Sentences

5.0.10 Result

By developing an ontology representation, sharing the subjective data for domain specific analysis can be done easily and human errors can be minimized and focus can be solely put on other research areas than manually labeling the dataset.

Chapter 6

CONCLUSION AND FUTURE WORK

Our approach to develop a framework for sentiment analysis in blogs was due to the fact that there is no system currently that classifies political blogs based on sentiments. We described novel ideas on developing filters by analyzing the sentence level rather than the document level and by including the custom developed pattern matching technique along with the machine learning technique and including the parts of speech tagging to further enhance the framework. We also describe the novel idea of multiple indexing to reduce the load on one index and thus improve the performance. We analyzed different machine learning approaches and presented a detailed description and analysis on each experiment. As explained, we feel that the best analysis for this framework is unigrams. We developed an ontology based representation for sentiments which would help other researchers. We also developed different applications, particularly the novel four panel view based on the label dataset, which presented a more clear picture on the sentiment analysis. The results that are obtained, demonstrate the potential for using this framework as a standard for different domains.

On the flip side, its known that developing domain specific patterns list and developing a domain specific training set data is not easy. The training set data should be available for researchers so that the focus can be on analyzing the sentences sentiments rather than

collection data for training the machine learning techniques. We need to further analyze this framework on different domains and develop a more generic list of patterns and training set data.

We have not handled sarcastic sentences, which has lot of potential for future work. We observed that while reading the entire blog, sarcastic sentences are perceived easily, but while analyzing the sentiments in the sentence level, they are hard to observe. Research have been going (Julius Quiaot 2007) on in this area and more focus in required in this field.

The nave Bayes method performs well, however, we are investigating on other machine learning techniques like Support Vector Machines and Maximum Entropy technique for enhancing the performance of our framework. For nave Bayes method, we are contemplating on expanding the ngram analysis on trigram technique to analyze the performance further.

We believe that combining the current framework with semantic orientation on the sentiments could yield better performance as shown in (Turney" 2002). Investigations in this area will yield superior results. We currently use multiple indices for indexing sentiment oriented sentences, which reduces load if a single indexing mechanism is used. Although this yields better results, computation power deteriorates, if there are more indices. Hence focus on combining the multiple indices in a way that if doesn't affect the performance is under examination. Analyzing trends in blogs have become a more sought after research topic for perceiving different analysis. Although we focused on this approach, we are currently planning to develop it further to detect the cold topics. The system can use the indexed metadata for analysis and since the framework is modular, plugging in different trend analysis would not be a problem. Hence our framework can be extended for analyzing different areas of research.

REFERENCES

[1] Akshay Java, Pranam Kolari, T. F. A. J. J. M., and Mayfield, J. 2007. "the blogvox opinion retrieval system". *Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006)*.

[2] Bo Pang, L. L., and Vaithyanathan, S. 2005. Movie review data - version 1.

[3] Chesley, P.; Vincent, B.; Xu, L.; and Srihari, R. K. 2006. Using verbs and adjectives to automatically classify blog sentiment. In *Proceedings of AAAI-CAAW-06, the Spring Symposia*.

[4] Choi, Y.; Cardie, C.; and Breck, E. Identifying expressions of opinion in context.

[5] Craig Macdonald, I. O. 2006. The trec blogs06 collection : Creating and analysing a blog test collection. 1–4.

[6] Das, S., and Chen, M. Yahoo! for amazon: Extracting market sentiment from stock message boards.

[7] Dave, K.; Lawrence, S.; and Pennock, D. M. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *World Wide Web*.

[8] Drezner, D. W., and Farrell", H. 2004. The power and politics of blogs.

[9] Engstrom, C. 2004. Topic dependence in sentiment classification. Master's thesis, University of Cambridge.

[10] Hatzivassiloglou, V., and Wiebe, J. 2000. Effects of adjective orientation and gradability on sentence subjectivity.

[11] Hearst, M. 1992. Direction-based text interpretation as an information access refinement.

[12] Hong Yu, V. H. Towards answering opinion questions: Separating facts from opinions.

[13] http://en.wikipedia.org/wiki/Blog. ”blogs.”.

[14] http://en.wikipedia.org/wiki/Confusion_matrix. Confusion matrix.

[15] http://en.wikipedia.org/wiki/Kullback Leibler_divergence. ”kullbackleibler divergence”.

[16] http://en.wikipedia.org/wiki/Social_media. ”social media.”.

[17] http://lucene.apache.org/java/docs. ”lucene”.

[18] http://tailrank.com. ”tailrank”.

[19] http://twitter.com. ”twitter”.

[20] http://wiki.java.net/bin/view/Javawsxml/Rome. ”rome parser”.

[21] http://www.alias i.com/lingpipe. ”lingpipe”.

[22] http://www.cs.cmu.edu/ cburch/words/top.html. ”top hundred words”.

[23] http://www.nypost.com/seven/05222007/news/nationalnews/mccain_sorry_for_outburst_nationalnews_charles_hurt.htm. ”mccain sorry for outburst”.

[24] Izzet Zorlu, S. S. 2005. Textmap: The entity search engine.

[25] Janyce Wiebe, E. R. 2005. Creating subjective and objective sentence classifiers from unannotated texts. *Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2005)*.

[26] Julius Quiaot, Hongcheng Mi, I.-H. M. 2007. Sentiment mining and indexing in opinmind. 1–2.

[27] Lada A. Adamic, N. G. 2005. The political blogosphere and the 2004 u.s. election: divided they blog. *Conference on Knowledge Discovery in Data Proceedings of the 3rd international workshop on Link discovery* 36 – 43.

[28] Mccallum, A., and Nigam, K. 2001. A comparison of event models for naive bayes text classification.

[29] Mullen, T., and Collier, N. 2004. Sentiment analysis using support vector machines with diverse information sources. *42nd Meeting of the Association for Computational Linguistics (ACL)*.

[30] Niblack, W., and Yi, J. Sentiment mining in webfountain. *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*.

[31] Pang, B., and Lee, L. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 271. Morristown, NJ, USA: Association for Computational Linguistics.

[32] Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs up? sentiment classification using machine learning techniques.

[33] Pranam Kolari, T. F., and Joshi, A. 2003. Svms for blogosphere: Blog identification and splog detection. In *In Proceedings of the AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*. AAAI Press.

[34] Pranam Kolari, A. J. T. F., and Joshi, A. 2006. Characterizing the splogosphere.

In *Proceedings with the 3rd annual workshop on weblogging Ecosystem: Aggregation, analysis and dynamics, 15th WWW Conference*.

[35] Riloff, E.; Wiebe, J.; and Wilson, T. 2003. Learning subjective nouns using extraction pattern bootstrapping.

[36] RSS. http://web.resource.org/rss/1.0.

[37] Theresa Wilson, Paul Hoffmann, J. W. 2005. Opinionfinder: a system for subjectivity analysis. 34–35.

[38] Turney", P. 2002. "thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews". *40th Annual Meeting of the Association for Computational Linguistics (ACL'02)* 417–424.

[39] Yi, J.; Nasukawa, T.; Bunescu, R.; and Niblack, W. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, 427. Washington, DC, USA: IEEE Computer Society.