Sentiment Analysis of Micro blogs using Opinion Mining Classification Algorithm

A. Tamilselvi[1], M. ParveenTaj[2]

[1]Research Scholar, Department of Computer Science,
Sri Jayendra Saraswathy Maha Vidyalaya College of Arts and Science, Coimbatore-5, India

[2]Research Supervisor, Department of Computer Science,
Sri Jayendra Saraswathy Maha Vidyalaya College of Arts and Science, Coimbatore-5, India

Abstract: ***Opinion mining and sentiment analysis is a fast growing topic with various world applications, from polls to advertisement placement. Traditionally individuals gather feedback from their friends or relatives before purchasing an item, but today the trend is to identify the opinions of a variety of individuals around the globe using micro blogging data. This paper discusses an approach where a publicized stream of tweets from the Twitter micro blogging site are preprocessed and classified based on their subjectivity word and semantic phrase content as positive, negative and irrelevant. Analyses the performance of various classifying algorithms based on their precision and recall in such cases. In this paper, focus on using Twitter, the most popular micro-blogging platform, for the task of sentiment analysis. Show how to automatically collect a corpus for sentiment analysis and opinion mining purposes. The classification of a review is predicted by the average semantic orientation of the phrases in the review that contain adjectives or adverbs. A phrase has a positive semantic orientation when it has good associations and a negative semantic orientation when it has bad associations this paper, the semantic orientation of a phrase is calculated as the mutual information between the given phrase and the word "excellent" minus the mutual information between the given phrase and the word "poor". Experimental evaluations show that proposed techniques are efficient and perform better than previously proposed methods.***

Keywords: opinion mining, part of speech, unigram, bigram.

1. Introduction

Micro blogging today has become a very popular communication tool among Internet users. Millions of users share opinions on different aspects of life every day. Therefore micro blogging web-sites are rich sources of data for opinion mining and sentiment analysis. Because micro blogging has appeared relatively recently, there are a few research works that were devoted to this topic. In our paper, we focus on using Twitter, the most popular micro blogging platform, for the task of sentiment analysis. We show how to automatically collect a corpus for sentiment analysis and opinion mining purposes. We perform linguistic analysis of the collected corpus and explain discovered phenomena. Using the corpus, we build a sentiment classifier that is able to determine positive, negative and neutral sentiments for a document. Experimental evaluations show that our proposed techniques are efficient and perform better than previously proposed methods. In our research, we worked with English; however, the proposed technique can be used with any other language.

Opinion mining (or sentiment analysis) has attracted great interest in recent years, both in academia and industry due to its potential applications. One of the most promising applications is analysis of opinions in social networks. Lots of people write their opinions in forums, micro blogging or review websites. This data is very useful for business companies, governments, and individuals, who want to track automatically attitudes and feelings in those sites. Namely, there is a lot of data available that contains much useful information, so it can be analyzed automatically. For instance, a customer who wants to buy a product usually searches the Web trying to find opinions of other customers or reviewers about this product. In fact, these kinds of reviews affect customer's decision.

2. Related Work

The population of blogs and social networks, opinion mining and sentiment analysis became a field of interest for many researches. In their survey, the authors describe existing techniques and approaches for an opinion-oriented information retrieval. However, not many researches in opinion mining considered blogs and even much less addressed micro blogging. The authors use web blogs to construct corpora for sentiment analysis and use emotion icons assigned to blog posts as indicators of users' mood. The authors applied SVM and CRF learners to classify sentiments at the sentence level and then investigated several strategies to determine the overall sentiment of the document. As the result, the winning strategy is defined by considering the sentiment of the last sentence of the document as the sentiment at the document level used emoticons such as ":-)" and ":- (" to form a training set for the sentiment classification. For this purpose, the author collected texts containing emoticons from Usenet newsgroups. The dataset was divided into "positive" (texts with happy emoticons) and "negative" (texts with sad or angry emoticons) samples. Emotion constrained classifiers: SVM and Navive Bayes were able to obtain up to 70% of accuracy on the test set.

The Affective Norms for English Words (ANEW) is being developed to provide a set of normative emotional ratings for a large number of words in the English language [1]. In this paper [2] we will describe a simple rule-based approach to automated learning of linguistic knowledge. This approach has been shown for a number of tasks to capture information in a clearer and more direct fashion without a compromise in performance. We describe a case study of a sentence-level categorization in which tagging instructions are developed and used by four judges to classify clauses from the Wall Street Journal as either subjective or objective[3]. Agreement among the four judges is analyzed, and, based on that analysis each clause is given a final classification. The major technical innovation is the use of a "maximum-entropy-inspired" model for conditioning and smoothing that let us successfully to test and combine many different conditioning events[4]. We also present some partial results showing the effects of different conditioning information, including a surprising 2% improvement due to guessing the lexical head's pre-terminal before guessing the lexical head from one or multiple documents into tight clusters. By placing highly related text units in the same cluster, SIMFINDER enables a subsequent content selection/generation component to reduce each cluster to a single sentence, either by extraction or by reformulation. We report on improvements in the similarity and clustering components of SIMFINDER [8], including a quantitative evaluation, and establish the generality of the approach by interfacing SIMFINDER to two very different summarization system.

Subjectivity [7] is a pragmatic, sentence-level feature that has important implications for text processing applications such as information extraction and information retrieval. We study the effects of dynamic adjectives, semantically oriented adjectives, and gradable adjectives on a simple subjectivity classifier, and establish that they are strong predictors of subjectivity.

This paper [10] presents a case study of analyzing and improving inter coder reliability in discourse tagging using statistical techniques. Bias-corrected tags are formulated and successfully used to guide a revision of the coding manual and develop an automatic classifier. This article shows that the density of subjectivity cluster [11]-[12] in the surrounding context strongly affects how likely it is that a word is subjective, and it provides the results of an annotation study assessing the subjectivity of sentences with high-density features. Finally, the clues are used to perform opinion piece recognition (a type of text categorization and genre detection) to demonstrate the utility of the knowledge acquired in this article.

3. Our Contribution

Our motivation for building the opinion detection and classification system described in this paper. Fully analyzing and classifying opinions involves tasks that relate to some fairly deep semantic and syntactic analysis of the text. These include not only recognizing that the text is subjective, but also determining who the holder of the opinion is, what the opinion is about, and which of many possible positions the holder of the opinion expresses regarding that subject. In this paper, we are presenting three of the components of our opinion detection and organization subsystem, which have already been integrated into our larger question-answering system. These components deal with the initial tasks of classifying articles as mostly subjective or objective, finding opinion sentences in both kinds of articles, and determining, in general terms and without reference to a specific subject, if the opinions are positive or negative.

The system identifies and uses this direct information in the following stages:

a) All conjunctions of adjectives are extracted from the corpus along with relevant morphological relations.
b) A log-linear regression model combines information from different conjunctions to determine if each two conjoined adjectives are of same or different orientation. The result is a graph with hypothesized same- or different-orientation links between adjectives.
c) The average frequencies in each group are compared and the group with the higher frequency is labeled as positive.

3.1. Data Preparation

Using Twitter API we collected a corpus of text posts and formed a dataset of three classes: positive sentiments, negative sentiments, and a set of objective texts (no sentiments). To collect negative and positive sentiments, we followed the same procedure. We queried Twitter for two types of emoticons: The two types of collected corpora will be used to train a classifier to recognize positive and negative sentiments. In order to collect a corpus of objective posts, we retrieved text messages from Twitter accounts of popular newspapers and magazines.

Stop words removal - The Natural Language Toolkit's (NLTK) stop word corpus for the English language to remove the stop words from the data. This helps eliminating the most common stop words from being included in the computation of n-grams and feature extraction.

Stemming - Twitter data is generally used with informal language and it includes internet jargons, slang and contemporary spellings. The very frugal in stemming so as to not risk truncating words and losing out on potential features. Employed basic stemming e.g. use of apostrophes.

Spelling correction - As Twitter users generally use informal language, there are often incorrect spellings in tweets. We used Jazzy Open Source Spell Checker to detect incorrect spellings in the tweets and replace them with the closest word from the English dictionary.

Emoticons mapping to sentiments - There are a multitude of emoticons that are used frequently in Twitter. We used an approach inspired by the method used and mapped some emoticons to positive and negative sentiments and discarded emoticons that are ambiguous or irrelevant to sentiments.

Filtering - Tweets contained a lot of metadata and quite a bit of noise which were removed. The following data was filtered, Identity numbers, date, time etc. of the tweets are Irrelevant tags, Hyperlinks, #tags e.g. #msnbc2012, Twitter handles e.g. @ pavanred, punctuation, special characters and digits are implemented.

3.2. Feature Extraction

In order to perform machine learning, it is necessary to extract certain clues from the text that may lead to an effective correct classification. Clues about the original data are usually stored in the form of a feature vector, F = (f1, f2…….. fn). Each coordinates of a feature vector represents one clue, also called a feature, “fi “of the original text.

On setting out to classify a document, we generally start with depicting a very large number of words that need to be considered, even though very few of the words in the corpus are actually expressing sentiment. These extra features have two clear drawbacks that need to be eliminated. The first is that they show down the process of document classification, since there are far more words than needed. The second is that they can actually reduce accuracy, since the classifier is obliged to consider these words when classifying a document.

Clearly, there is an advantage in using fewer features; so in order to remove some of the unnecessary features, we resort to feature selection. As the name suggests, feature selection is a process through which we run across the corpus before the classifier has been trained and remove any features that seem unnecessary. This allows the classifier to fit a model to the problem more quickly as there will be less information to consider, thus allowing it to classify items faster.

N-gram features - N-grams are capable of capturing context to some extent and are widely used in Natural Language Processing tasks. Whether higher order n-grams are useful is a matter of debate. It has been reported by researchers that unigrams outperform bigrams when classifying movie reviews by sentiment polarity, but other researchers found that in some settings, bigrams and trigrams perform better. The processed data is used to extract features that will be used to train our classifier. We have experimented with unigrams, bigrams, trigrams and combination of unigrams and bigrams. The data was tokenized by spaces using NLTK and these tokens were subject to NLTK to generate n-grams.

Part of Speech features-Parts of Speech information is most commonly exploited in all NLP tasks. One of the most important reasons is that they provide a crude form of word sense disambiguation. Since the language used in Twitter is generally informal, part of speech tagging isn’t very accurate for tweets. We used both NLTK Part Of Speech tagger and Open NLP Part Of Speech tagger along with a heuristic that adjectives and/or adverbs, JJ, JJR, JJS, RB, RBR and RBS in the Penn Tree bank target, are generally used to articulate opinions in natural language. So, we further process the data by excluding all data except the entity, adjectives, adverbs and words such as not, couldn’t etc which generally indicate a reversal of sentiment. This is similar to the way opinion reversing words. After using part of speech taggers, we experimented with unigrams, bigrams and combination of unigrams and bigrams. Experimented with term frequency-inverse document frequency (tf - idf) where we considered only the most frequent terms ordered by tf - idf. We used the absolute approach of considering all the n-grams as features as well. Part-of-speech tagging is harder than just having a list of words and their parts of speech, because some words can represent more than one part of speech at different times, and because some parts of speech are complex or unspoken. This is not natural languages (as opposed to many artificial languages), a large percentage of word-forms are ambiguous. For example, even "dogs", which is usually thought of as just a plural noun, can also be a verb: The sailor dogs the barmaid.

3.3. Semantically Oriented Words

Having distinguished whether a sentence is a fact or opinion, we separate positive, negative, and neutral opinions into three classes. We base this decision on the number and strength of semantically oriented words (either positive or negative) in the sentence. We first discuss how such words are automatically found by our system, and then describe the method by which we aggregate this information across the sentence.

To determine which words are semantically oriented [9], in what direction, and the strength of their orientation, we measured their co-occurrence with words from a known seed set of semantically oriented words. The approach is based on the hypothesis that positive words co-occur more than expected by chance, and so do negative words; this hypothesis was validated, at least for strong positive/negative words. As seed words, we used subsets of the 1,336 adjectives that were manually classified as positive (657) or negative (679). In earlier work only singletons were used as seed words; varying their number allows us to test whether multiple seed words have a positive effect in detection performance. We experimented with seed sets containing 1, 20, 100 and over 600 positive and negative pairs of adjectives. For a given seed set size, we denote the set of positive seeds as ADJ p and the set of negative seeds as ADJ n. We then calculate a modified log-likelihood ratio L(Wi, POSj) for a Wi with part of speech POSj (j can be adjective, adverb, noun or verb) as the ratio of its collocation frequency with ADJp and ADJ n within sentence,

$$L(W_i, POS_j = \log\left(\frac{\frac{\mathrm{Freq(Wi,POSj,ADJp)}+\varepsilon}{\mathrm{Freq(Wall,POSj,ADJp)}}}{\frac{\mathrm{Freq(Wi,POSj,ADJn)}+\varepsilon}{\mathrm{Freq(Wall,POSj,ADJn)}}}\right)$$

Where Freq (Wall, POSj, ADJp) represents the collocation frequency of all words Wall of part of speech POSj with ADJp and∈ is a smoothing constant, We used Brill’s tagger (Brill, 1995) to obtain part-of-speech information.

3.4 Sentence Polarity Tagging

As our measure of semantic orientation [5] across an entire sentence we used the average per word log likelihood scores defined in the preceding section. To determine the orientation of an opinion sentence, all that remains is to specify cutoffs tp and tn so that sentences for which the average log-likelihood score exceeds tp are classified as positive opinions, sentences with scores lower than tn are classified as negative opinions, and sentences with in-between scores are treated as neutral opinions. Optimal values for tp and tn are obtained from the training data via density estimation using a small, hand-labeled subset of sentences we estimate the proportion of sentences that are positive or negative. The values of the average log-likelihood score that correspond to the appropriate tails of the score distribution are then determined via Monte Carlo analysis of a much larger sample of unlabeled training data.

4. Sentiment Classifier

4.1 Bayesian Opinion Mining

Bayesian classifiers are based around the Bayes rule, a way of looking at conditional probabilities that allows you to flip the condition around in a convenient way. A conditional probably is a probably that event X will occur, given the evidence Y. That is normally written P(X | Y). The Bayes rule allows us to determine this probability when all we have is the probability of the opposite result and of the two components individually: P(X | Y) = P(X) P(Y | X) / P(Y). This restatement can be very helpful when we're trying to estimate the probability of something based on examples of it occurring.

Formula looks like this.

$$\mathrm{P(Sentiment|Sentence} = \frac{\mathrm{P(Sentiment)P(Sentence|Sentiment)}}{\mathrm{P(Sentence)}}$$

In this case, we're trying to estimate the probability that a document is positive or negative, given its contents. We can restate that so that is in terms of the probability of that document occurring if it has been predetermined to be positive or negative. This is convenient, because we have examples of positive and negative opinions from our data set above. The thing that makes this a "naive" Bayesian process is that we make a big assumption about how we can calculate at the probability of the document occurring: that it is equal to the product of the probabilities of each word within it occurring. This implies that there is no link between one word and another word. This independence assumption is clearly not true: there are lots of words which occur together more frequently that either does individually, or with other words, but this convenient fiction massively simplifies things for us, and makes it straightforward to build a classifier.

We can estimate the probability of a word occurring given a positive or negative sentiment by looking through a series of examples of positive and negative sentiments and counting how often it occurs in each class. This is what makes this supervised learning - the requirement for pre-classified examples to train on. We can drop the dividing P(line), as it's the same for both classes, and we just want to rank them rather than calculate a precise probability. We can use the independence assumption to let us treat P (sentence | sentiment) as the product of P (token | sentiment) across all the tokens in the sentence.

So, we estimate P (token | sentiment) as

Count (this token in class) + 1 / count (all tokens in class) + count (all tokens)

The extra 1 and count of all tokens is called 'add one' or Laplace smoothing, and stops a 0 finding its way into the multiplications. If we didn't have it any sentence with an unseen token in it would score zero. The classify function starts by calculating the prior probability (the chance of it being one or the other before any tokens are looked at) based on the number of positive and negative examples - in this example that'll always be 0.5 as we have the same amount of data for each. We then tokenize the incoming document, and for each class multiply together the likelihood of each word being seen in that class. We sort the final result, and return the highest scoring class.

The Bayes Naïve Classifier selects the most likely classification Vnb given the attribute values a1, a2…….an.

This results in: Where:

n= the number of training examples for which v = vj
ne = number of examples for which v = vj and a=ai
p = a priori estimate for P(ai | vj)
m = the equivalent sample size

A naive Bayes classifier assumes that the presence of a particular feature of a class is unrelated to the presence of any other feature. For example, a person may be considered to be a male if he is tall, has short hair and a strong build. Even if these features depend on each other or upon the existence of the other features, a naïve Bayes classifier considers all of these properties to independently contribute to the probability that the person is a male.

4.2 Turney Opinion Mining

Step 1: Part-of-speech (POS) tagging Extracting two consecutive words (two word phrases) from reviews if their tags conform to some given patterns, e.g., (1) JJ, (2) NN.

Step 2: Estimate the sentiment orientation (SO) of the extracted phrases Pwords = a set of words with positive semantic orientation Nwords = a set of words with negative semantic orientation A (word1, word2) = a measure of association between word1 and word2

$$SO-A(word) = \sum_{pword\in Pwords} A(word, pword) - \sum_{nword\in Nwords} A(Word, nword)$$

Pwords = {good, nice, excellent, positive, fortunate, correct, and superior}
Nwords = {bad, nasty, poor, negative, unfortunate, wrong, and inferior}.

Positive Review:

Example: love the local branch however communication may break down if they have to go through head office.

Avg. SO Value=0.1414

Negative Review:

Example: Do not bank here; their website is even worse than their actual locations. Avg.So value: -0.0766

Steps3: Point wise Mutual Information (PMI), The Point wise Mutual Information (PMI) between two words, word1 and word2, is defined

$$\mathrm{PMI}(\mathrm{word1}, \mathrm{word2}) = \log_2 \frac{\mathrm{p}(\mathrm{word1}\& \, word2)}{\mathrm{p}(\mathrm{word1})\mathrm{p}(\mathrm{word2})}$$

Here, p(word1 & word2) is the probability that word1 and word2 co-occur. If the words are statistically independent, the probability that they co-occur is given by the product p(word1) p(word2). The ratio between p(word1 & word2) and p(word1) p(word2) is a measure of the degree of statistical dependence between the words. The first step of the algorithm is to extract phrases containing adjectives or adverbs. Past work has demonstrated that adjectives are good indicators of subjective, evaluative sentences.

The Semantic Orientation (SO) of a phrase, *phrase*, is calculated here as follows:

SO(*phrase*) = PMI(*phrase*, "excellent")
PMI(*phrase*, "poor")

The reference words "excellent" and "poor" were chosen because, in the five star review rating system, it is common to define one star as "poor" and five stars as "excellent". SO is positive when *phrase* is more strongly associated with "excellent" and negative when *phrase* is more strongly associated with "poor".

PMI-IR estimates PMI by issuing queries to a search engine (hence the IR in PMI-IR) and noting the number of hits (matching documents). The following experiments use the AltaVista Advanced Search engine5, which indexes approximately 350 million web pages (counting only those pages that are in English). I chose AltaVista because it has a NEAR operator. The AltaVista NEAR operator constrains the search to documents that contain the words within ten words of one another, in either order. Previous work has shown that NEAR performs better than AND when measuring the strength of semantic association between words.

Let hits (*query*) are the number of hits returned, given the query. The following estimate of SO can be derived from equations (1) and (2) with some minor algebraic manipulation, if co occurrence is interpreted as NEAR:

$$\mathrm{SO}\,(phrase) = \log 2 \left[\frac{\mathrm{hits}(phrase \text{ NEAR ``excellent''})\ \mathrm{hits}(\text{``poor''})}{\mathrm{hits}(phrase \text{ NEAR ``poor''})\ \mathrm{hits}(\text{``excellent''})}\right]$$

is a log-odds ratio. To avoid division by zero, I added 0.01 to the hits. I also skipped *phrase* when both hits(*phrase* NEAR "excellent") and hits(*phrase* NEAR "poor") were (simultaneously) less than four. These numbers (0.01 and 4) were arbitrarily chosen. To eliminate any possible influence from the testing data, I added "AND (NOT host: opinions)" to every query, which tells AltaVista not to include the Opinions Web site in its searches.

5. Results

To the best of our knowledge, there is no annotated dataset for opinion retrieval in Twitter. Therefore, we created a new dataset for this task2. We crawled and indexed about 30 million tweets using the Twitter API. All tweets are English. Using these tweets we implemented a search engine. Seven people (a woman and six men) were asked to use our search engine. They were allowed to post any query. Given a query the search engine would present a list of 100 tweets ranked based on the BM25 score. Based on the principle about the tweet whether expresses opinion about a given query, people assigned a binary label to every tweet. Finally we totally collected 50 queries and all judged tweets. The average query length was 1.94 words and the average number of relevant tweets per query was 16.62.

It suggests some social information can indeed help opinion retrieval in Twitter. We see that the URL feature is the most effective feature, perhaps because most textual content in these tweets are objective introductions. Also, spammers usually post tweets including links and features dealing with links might help reduce spam. The effect of URL, Statuses and Followers features for tweets ranking also supports our approach of using social information and structural information to generate "pseudo" objective tweets. We examined the impact of the dataset size on the performance of the system. To measure the performance, we use F-measure

$$F = (1 + \beta^2)\frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{recall} + \text{recall}}$$

Compute accuracy of the classifier on the whole evaluation dataset, i.e.:

$$\text{accuracy} = \frac{\mathrm{N}(\text{correct classifications})}{\mathrm{N}(\text{all classifications})}$$

We measure the accuracy across the classifier's decision

$$\text{decision} = \frac{\mathrm{N}(\text{retrieved documents})}{\mathrm{N}(\text{all documents})}$$

Table.1: Table showing user query: Barack Obama

Sentiment	*Precision*	*Recall*	*f-measure*
Positive	37.8	30.5	33.8
Negative	33.9	60.4	43.4

Table.2: Table showing user query: IPL2013

Sentiment	*precision*	*recall*	*f-measure*
Positive	49.0	75.8	60.2
Negative	44.9	21.1	28.7

Table.3: Table showing comparison of classification algorithm

Classifiers	*Avg. Accuracy*	*Max. Accuracy*	*Avg F*
SVM	69.82%	71.33%	0.688
Mixed Algorithm	73.25%	77.60%	0.728
Naïve Bayes (NB)	80.92%	84.58%	0.809
Turney	91.86%	94.82%	0.920

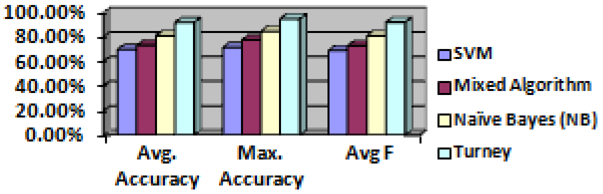

Figure 1: Comparison of classification algorithm

Using sentiment-topic features consistently performs better than using semantic features. With as few as 500 features, augmenting the original feature space with sentiment-topics already achieves 80.2% accuracy. Although with all the features included, NB trained with semantic features performs better than that with sentiment-topic features, we can still draw a conclusion that sentiment-topic features should be preferred over semantic features for the sentiment classification task since it gives much better results with far less features.

Table 4: Cross comparison results of all the features

Dataset	*Feature*	*Positive Sentiment*	*Negative Sentiment*	*Average*
Twitter	Unigrams	82.20	79.30	80.75
	POS	83.70	79.50	81.60

The Semantic approach outperforms the Unigrams and POS baselines in all categories and for all three datasets. However, For example, the semantic approach produced higher P, R, and F1 for the twitter dataset, with F1 4.4% higher than Unigrams, 3.5% higher than POS, and 2.1% higher than the sentiment-topic features. Unigrams and POS baselines, with 5.2% and 2.4% higher F1 respectively. However, in the Twitter dataset, F1 from semantic features was 0.4% lower than from the topic model, although Precision was actually higher by 1.7%.

Table 5: Grams comparison results of all the features

Algorithm	*Unigrams*	*Bi-grams*	*Jointly*
Bayes	0.73	0.72	0.72
Turney	0.81	0.76	0.82

This table **a**n algorithm for sentiment analysis of Twitter messages was developed as part of the social network investigation system. One of the modules of this system is the sentiment module for text messages.

6. Conclusion

The expression of opinions of users in specialized sites for evaluation of products and services, and also on social networking platforms, has become one of the main ways of communication, due to spectacular development of web environment in recent years. The large amount of information on these platforms make them viable for use as data sources, in applications based on opinion mining and sentiment analysis. This paper discusses an approach where a stream of tweets from the Twitter micro blogging site are preprocessed and classified based on their feature content as positive, negative and irrelevant; and analyses the performance of various classifying algorithms based on their precision and recall in such cases. The limitations of this work include the time required for queries and, for some applications, the level of accuracy that was achieved. The former difficulty will be eliminated by progress in hardware. The latter difficulty might be addressed by using semantic orientation combined with other features in a supervised classification algorithm.

7. Future Work

There is scope for lot of work further in political opinion mining of tweets. Future work can involve:

- User specific sentiment inference, since political sentiments of most people do not change often, we can identify sentiments using superlative adjectives/adverbs to infer the sentiment of a particular user in case of ambiguity such as a comparative adjective/adverb involving multiple entities.
- To experiment with semantics, that is to create triples involving entities, users, sentiments
- linguistic and contextual clues: the development of the application described here is a first stage towards a more complete system, and also context the work within a wider framework of social media monitoring which can lead to interesting new perspectives when combined with relevant research in related areas such as trust, archiving and digital libraries.
- Further, providing drill-down and roll-up services (i.e., polarity analysis in a country, in regions, in states etc.) are also targets of future work.
- This work can be extended through incorporation of better spell correction mechanisms (may be at phonetic level) and word sense disambiguation. Also we can identify the target and entities in the tweet and the orientation of the user towards them

References

[1] M. M. Bradley and P. J. Lang. 1999. Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings. Technical Report C-1, the Center for Research in Psychophysiology, University of Florida, Gainesville, Florida.

[2] Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. Computational Linguistics.

[3] Rebecca Bruce and Janyce Wiebe. 1999. Recognizing subjectivity: A case study in manual tagging. Natural Language Engineering, 5(2).

[4] Eugene Charniak. 2000. A maximum-entropy–inspired parser. In Proceedings of NAACL-2000.

[5] Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In Proceedings of the 35th Annual Meeting of the ACL and the 8th Conference of the European Chapter of the ACL, pages 174–181, Madrid, Spain, July. Association for Computational Linguistics.

[6] Vasileios Hatzivassiloglou and Janyce Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity. In Conference on Computational Linguistics (COLING-2000).

[7] Vasileios Hatzivassiloglou, Judith Klavans, Melissa Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen McKeown. 2001. SIMFINDER: A flexible

clustering tool for summarization. In Proceedings of the Workshop on Summarization in NAACL-01.

[8] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumps up? Sentiment classification using machine learning techniques. In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP-02), Philadelphia, Pennsylvania.

[9] Peter Turney. 2002. Thumps up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania.

[10] Janyce Wiebe, Rebecca Bruce, and Thomas O'Hara. 1999. Development and use of a gold standard data set for subjectivity classifications. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99), pages 246–253.

[11] Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and M. Martin. 2002. Learning subjective language. Technical Report TR-02-100, Department of Computer Science, University of Pittsburgh, Pittsburgh, Pennsylvania.

[12] Janyce Wiebe. 2000. Learning subjective adjectives from corpora. In Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000), Austin, Texas.

Author Profile

A. Tamilselvi M. Phil (Computer Science) Research Scholar, Sri Jayendra Saraswathy Maha VidyalayaCollege of Arts and Science, His areas of Interest are Networking, Data Mining, and opinion mining.

M. ParveenTaj has done M.C.A., ADCA and M. Phil. She is Research Supervisor, Department of Computer Science, Sri Jayendra Saraswathy Maha Vidyalaya College of Arts and Science, Coimbatore-5, India. Her area of interest is Networking, Data mining, and Software Engineering. She has presented two papers in an International level