Sentiment Analysis of Movie Reviews using POS tags and Term Frequencies

Oaindrila Das
IIIT Bhubaneswar
Bhubaneswar
Orissa, India

Rakesh Chandra Balabantaray
IIIT Bhubaneswar
Bhubaneswar
Orissa, India

ABSTRACT

Sentiment analysis and opinion mining play an important role in judging and predicting people's views. Recently, sentiment analysis has focused on assigning positive and negative polarities to opinions. More methods are being devised to find the weightage of a particular expression in a sentence, whether the particular expression gives the sentence a positive, negative or a neutral meaning. Most of the work on sentiment analysis in the past has been carried out by determining the strength of a subjective expression within a sentence using the parts of speech. Sentiment analysis tries to classify opinion sentences in a document on the basis of their polarity as positive or negative, which can be used in various ways and in many applications for example, marketing and contextual advertising, suggestion systems based on the user likes and ratings, recommendation systems etc. This paper presents a novel approach for classification of online movie reviews using parts of speech and machine learning algorithms.

General Terms

Natural Language Processing, Data Mining

Keywords

bigrams, POS tagger, sentiment analysis, SVM lite, Weka

1. INTRODUCTION

It is human nature to consult their dear ones when it comes to decision making. Most of their decisions in the real world are influenced by the thinking that, how other people would perceive their decision. Just a decade ago, when internet was not so popular, people used to take decisions about buying certain service or product based on their friends' or critic's recommendation. However, the amount of information available for decision making was limited. But with the popularity of internet, the big-data explosion and the ability of the people to learn and exploit the web has made tremendous amount of information available, which can be used to objectively make important decisions. Owing to the ease of internet use coupled with the numerous sources of information people have started referring to the internet for making decisions. These decisions comprise of seeking an opinion on brands, products, services, religion, politics, economics, entertainment etc. The advent of user interaction platforms such as blogs, forums, peer to peer networks and various other types of social media has given the consumers an unbounded reach and power to share their brand experiences and opinions, positive or negative, regarding any product or service. However, this information explosion on the web might be misleading at times because there are opinions put forward in dissimilar manners by different consumers. Thus, there is an urgent need to find a mechanism that can process this unstructured information in a better way and represent the associated sentiments more objectively.

Approximately one lakh blogs are created every day, which roughly add up to around one million posts. Majority of these posts are related to opinions given by customers on products and services. As a result, company owners and businessmen have realized that consumer opinions available on the web can significantly influence the decisions taken by other customers thereby affecting their purchase decisions and brand values. This trend of customer opinions posted online has made the company owners and businessmen focus on the customer sentiments and respond to them through social media, customer care; thereby changing the quality of their product and designing novel marketing strategies to get an improved customer feedback.

The change in the customer choice influenced by different opinions, has outsmarted the traditional opinion monitoring methods and these methods have been left behind because of the fast growing information available through various media. The process of decision making has improved as compared to the traditional methods since the advent of internet. Thus, the fields of sentiment analysis has a very important role to play in future to objectively categorize this information and improve the decision making process. Sentiment analysis or opinion mining is the computational study of opinions, sentiments and emotions expressed in text. [1]It is a challenging natural language processing or text mining problem. Due to its tremendous value for practical application, there has been an explosive growth of both research in academia and applications in the industry.

2. RELATED WORK

Pang et al. (Pang et al., 2002) [2] used the traditional n-gram approach along with POS information as a feature to perform machine learning for determining the polarity. They used Naive Bayes Classification, Maximum Entropy and Support Vector Machines on a threefold cross validation. In their experiment, they tried different variations of n-gram approach like unigrams presence, unigrams with frequency, unigrams + bigrams, bigrams, unigrams+ POS, adjectives, most frequent unigrams, unigrams + positions. They concluded from their work that incorporating the frequency of matched n-gram might be a feature which could decay the accuracy. Maximum accuracy achieved by them among all the experiments they performed was 82.9% which was obtained in unigrams presence approach on SVM.

Turney (2004) [3] worked on POS information. He used some tag patterns with a window of maximum three words that is till trigrams. In his experiments, he considered JJ(adjective), RB(adverb), NN(single common nouns), NNS(plural common nouns) POS-tags with some set of rules for classification. His work is extension to the work done on adjectives alone (Hatzivassiloglou and McKeown, 2004) because he considers RB, NN/NNS. Given a phrase he calculates the PMI (Point-wise Mutual Information) from the strong positive word “excellent” and also from the strong negative word “poor” and the difference will give you the semantic orientation of the phrase. The first step of the algorithm is to extract phrases containing adjectives or adverbs. The algorithm extracts two consecutive words, where one member of the pair is an adjective or an adverb and the other provides context. The second step is to estimate the semantic orientation of the extracted phrases, using the PMI-IR algorithm. This algorithm uses mutual information as a measure of the strength of semantic association between two words. The third step is to calculate the average semantic orientation of the phrases in the given review and classify the review as recommended if the average is positive and otherwise not recommended.

Kim and Hovy [4] describe an opinion as a quadruple [Topic, Holder, Claim, Sentiment] in which the Holder believes a Claim about the Topic, and in many cases associates a Sentiment, such as good or bad, with the belief. Given a topic and a set of texts, the system operates in four steps. First it selects sentences that contain both the topic phrase and holder candidates. Next, the holder-based regions of opinion are delimited. Then the sentence sentiment classifier calculates the polarity of all sentiment-bearing words individually. Finally, the system combines them to produce the holder's sentiment for the whole sentence.

Pang and Lee (2004) [5] proposed a novel machine learning method that applies text categorization techniques only for the subjective portions of the document. For extracting the subjective portions they have used an efficient technique for finding minimum cuts in graphs. They propose the following two step process: labeling the sentences in the document as either subjective or objective, discarding the objective sentences; applying a standard machine learning classifier to the resulting extract of subjective sentences. This step prevents the polarity classifier from considering any irrelevant text. The subjectivity extracts that they create accurately represent the sentiment information of the originating documents in a much more concise form. They have used subjectivity extraction methods based on minimum cuts formulation which provides an efficient, intuitive and effective means for integrating inter-sentence level contextual information. They have used a subjectivity detector that determines whether each sentence is subjective or not; by discarding the objective ones thus creating an extract containing only the subjective content of a review. This extract is then given to a document polarity classifier.

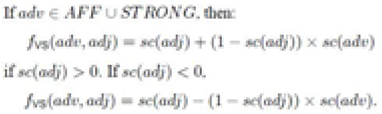
Benamara et al. (2007) [6] classify adverbs of degree into the following categories:

- Adverbs of affirmation (AFF): these include adverbs such as absolutely, certainly, exactly, totally, and so on.
- Adverbs of doubt (DOUBT): these include adverbs such as possibly, roughly, apparently, seemingly, and so on.
- Strong intensifying adverbs (STRONG): these include adverbs such as astronomically, exceedingly, extremely, immensely, and so on.
- Weak intensifying adverbs (WEAK): these include adverbs such as barely, scarcely, weakly, slightly, and so on.
- Negation and Minimizers (NEG): these include adverbs such as "hardly". We treat these somewhat differently than the preceding four categories as they usually negate sentiments.

They propose three alternative algorithms (different f's) to assign a score to a unary AAC (adverb adjective combination). All three algorithms can be extended to apply to binary AAC and negated AAC.

- Variable scoring

Suppose adj is an adjective and adv is an adverb. The variable scoring method (VS) works as shown in Figure.1 and Figure.2.



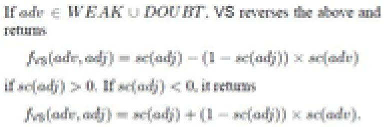
If $adv \in AFF \cup STRONG$, then:

$$f_{VS}(adv, adj) = sc(adj) + (1 - sc(adj)) \times sc(adv)$$

if $sc(adj) > 0$. If $sc(adj) < 0$,

$$f_{VS}(adv, adj) = sc(adj) - (1 - sc(adj)) \times sc(adv).$$

Fig:1 variable scoring



If $adv \in WEAK \cup DOUBT$, VS reverses the above and returns

$$f_{VS}(adv, adj) = sc(adj) - (1 - sc(adj)) \times sc(adv)$$

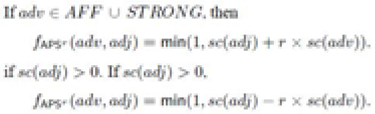
if $sc(adj) > 0$. If $sc(adj) < 0$, it returns

$$f_{VS}(adv, adj) = sc(adj) + (1 - sc(adj)) \times sc(adv).$$

Fig:2 variable scoring

- Adjective priority scoring

In variable scoring, the weight with which an adverb is considered depends upon the score of the adjective that it is associated with. In contrast, in Adjective Priority Scoring (APS), they select a weight between zero to one. This weight denotes the importance of an adverb in comparison to an adjective that it modifies. Weight can vary based on different criteria. The adjective priority scoring method works as shown in Figure.3 and Figure.4.



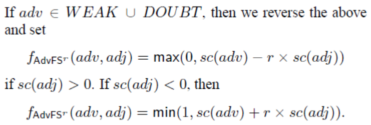
If $adv \in AFF \cup STRONG$, then

$$f_{\mathsf{APS}^r}(adv, adj) = \min(1, sc(adj) + r \times sc(adv)).$$

if $sc(adj) > 0$. If $sc(adj) > 0$,

$$f_{\mathsf{APS}^r}(adv, adj) = \min(1, sc(adj) - r \times sc(adv)).$$

Fig:3 Adjective priority scoring



If $adv \in WEAK \cup DOUBT$, then we reverse the above and set

$$f_{\mathsf{AdvFS}^r}(adv, adj) = \max(0, sc(adv) - r \times sc(adj))$$

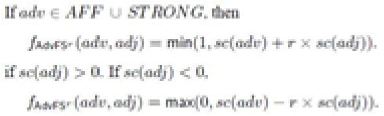
if $sc(adj) > 0$. If $sc(adj) < 0$, then

$$f_{\mathsf{AdvFS}^r}(adv, adj) = \min(1, sc(adv) + r \times sc(adj)).$$

Fig:4 Adjective priority scoring

- Adverb first scoring

In this algorithm, they take the complementary view that instead of weighting the adverb, they should modify the adverb score by weighting the adjective score using a term that measures the weight of an adjective's importance in an AAC, relative to the importance of the adverb. The adverb first scoring method works as shown in Figure.5 and Figure.6.



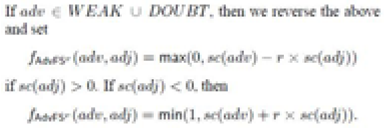
If $adv \in AFF \cup STRONG$, then

$$f_{\mathsf{AdvFS}^r}(adv, adj) = \min(1, sc(adv) + r \times sc(adj)).$$

if $sc(adj) > 0$. If $sc(adj) < 0$,

$$f_{\mathsf{AdvFS}^r}(adv, adj) = \max(0, sc(adv) - r \times sc(adj)).$$

Fig:5 Adverb first scoring



If $adv \in WEAK \cup DOUBT$, then we reverse the above and set

$$f_{\mathsf{AdvFS}^r}(adv, adj) = \max(0, sc(adv) - r \times sc(adj))$$

if $sc(adj) > 0$. If $sc(adj) < 0$, then

$$f_{\mathsf{AdvFS}^r}(adv, adj) = \min(1, sc(adv) + r \times sc(adj)).$$

Fig:6 Adverb first scoring

They have used an algorithm for scoring the strength of sentiment on a topic t in a document d which is as follows.

1. Let Rel(t) be the set of all sentences in d that directly or indirectly reference the topic t.

2. For each sentence s in Rel(t), let Appl+(s) (resp. Appl¡(s)) be the multiset of all AACs occurring in s that are positively(resp. negatively) applicable to topic t.

3. Return strength as per the formula shown in Figure.7

$$\frac{\Sigma_{s\in Rel(t)}\Sigma_{a\in Appl+(s)} score(a) - \Sigma_{s\in Rel(t)}\Sigma_{a'\in Appl-(s)} score(a')}{card(Rel(t))}.$$

Fig:7 Formula

The first step uses well known algorithms to identify sentences that directly or indirectly reference a topic, while the second step finds the AACs applicable to a given topic by parsing it in a straightforward manner. The third step is key: it says that we classify the applicable AACs into positive and negative ones. We sum the scores of all applicable positive AACs and subtract from it, the sum of scores of all applicable negative AACs. We then divide this by the number of sentences in the document to obtain an average strength of sentiment measure.

Blitzer et al., (2007) [7] hover into the field of domain adaptation for sentiment classifiers using product reviews available online. They extend the structural correspondence learning algorithm to sentiment classification. This algorithm reduces the relative error that is caused due to the adaptation between domains. They also identify a domain similarity measure that gives the potential of a sentiment classifier to adapt from one domain to another. Their algorithm selects labeled and unlabeled data from source and target domains. Using the SCL algorithm they choose a certain number say *m* pivot features occurring frequently in both the domains. Then the algorithm models the correlation between the selected pivot features and the other features by training linear pivot predictors to predict the occurrences of each pivot in the unlabeled data from both the domains. The efficiency of the SCL algorithm depends on the way the pivot features are chosen. For sentiment classification it is vital that the pivot features chosen are good predictors of the source label. Therefore they had chosen the words having highest mutual information with the source labels.

Bakliwal et al. (2009) [8] devised a new scoring function and tested on two different approaches which are as follows.

- Simple Ngram matching:

Unigrams, bigrams and trigrams of a review are been used to assign score to a review and thus classify it as positive or negative.

- POS tagged Ngram matching:

Ngrams in this case are formed using the POS tagged information of a review, trigrams, bigrams and unigrams combination of only adjectives and adverbs are used for scoring review.

To perform polarity classification they devised their own algorithm. This algorithm was applied on all approaches. In their experiments they performed 5-fold cross-validation and they divided the pre-annotated data into two parts namely training set and testing set to check the correctness. After dividing the data they form trigrams, bigrams and unigrams on the training data and store them in individual n-gram dictionary. They create two separate models each for positive and negative polarity. For every testing review they create trigrams in the similar manner. Then they check if this trigram exists in their positive and negative trigram dictionary. If it exists then, they increase the count of trigram matched else they break this trigram into two bigrams. These bigrams thus formed are cross checked in the bigram dictionary, if found then the bigram match count is increased otherwise each bigram is further split into two unigrams. These unigrams are then checked against the unigram dictionary. They propose a scoring function which gives priority to trigram matching followed by bigrams and unigrams. Figure. 8 shows the diagrammatic representation of the algorithm.

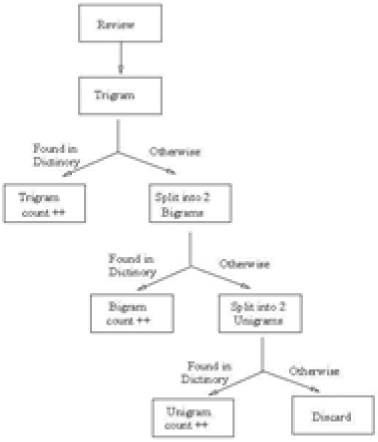


Fig:8 Algorithm flow

Dmitriy Bespalov et al. [9] propose an efficient embedding for modeling higher-order (n-gram) phrases that projects the n-grams to low-dimensional latent semantic space, where a classification function can be defined. They utilize a deep neural network to build a unified discriminative framework that allows for estimating the parameters of the latent space as well as the classification function with a bias for the target classification task at hand. They apply the framework to large-scale sentimental classification task. They evaluate the performance of the proposed method on two large data sets for online product reviews.

Wang Hao et al., (2012) [10] designed a system for sentiment analysis of twitter sentiments that were tweeted during the 2012 US presidential elections. The design of the sentiment model used in their system was based on the assumption that the opinions expressed would be highly subjective and contextualized. Therefore, for generating data for model training and testing, they used a crowdsourcing approach to do sentiment annotation on in-domain political data. To create a baseline sentiment model, they used Amazon Mechanical Turk (AMT) to get as varied a population of annotators as possible. They designed an interface that allowed annotators to perform the annotations outside of AMT so that they could participate anonymously. The Turkers were asked their age, gender, and to describe their political orientation. Then they were shown a series of tweets and asked to annotate the tweets' sentiment (positive, negative, neutral, or unsure), whether the tweet was sarcastic or humorous, the sentiment on a scale from positive to negative, and the tweet author's political orientation on a slider scale from conservative to liberal. Their sentiment model is based on the sentiment label and the sarcasm and humor labels. Their training data consists of nearly 17000 tweets (16% positive, 56% negative, 18% neutral, 10% unsure), including nearly 2000 that were multiply annotated to calculate inter-annotator agreement. About 800 Turkers contributed to their annotation. The statistical classifier they use for sentiment analysis is a naïve Bayes model on unigram features. Their features are calculated from tokenization of the tweets that attempts to preserve punctuation that may signify sentiment (e.g., emoticons and exclamation points) as well as twitter specific phenomena (e.g., extracting intact URLs). Based on the data they collected, their classifier performs at 59% accuracy on the four category classification of negative, positive, neutral, or unsure. These results exceed the baseline of classifying all the data as negative, the most prevalent sentiment category (56%). The choice of their model was not strictly motivated by global accuracy, but took into account class-wise performance so that the model performed well on each sentiment category.

Mostafa Mohamed, (2013) [11] used an expert-predefined lexicon including around 6800 seed adjectives with known orientation to conduct the analysis of consumer brand sentiments. Their results indicate a generally positive consumer sentiment towards several famous brands. By using both a qualitative and quantitative methodology to analyze brands' tweets, their study adds breadth and depth to the debate over attitudes towards cosmopolitan brands. Since Twitter is the most large, popular and well-known micro blog Web site, it was selected to conduct the analysis reported in their study. The data used represent a random set of Twitter posts from July 18, 2012, to August 17, 2012. The data comprised 3516 tweets for sixteen brands. To guarantee representativeness, sample selection had been varied by day of the week and hours in the day. Categorizing words for sentiment analysis was a major step in applying the technique. Broadly speaking, there are two widely used methods for sentiment orientation identification: the lexicon-based approach and the corpus-based method. However, since the corpus-based method has rarely been used to analyze sentiment orientation, they focused here on the lexicon-based method. They used the twitteR, the plyr, stringr and the ggplot2 libraries in the R software package version 2.15 to conduct the quantitative sentiment score. Finally, they used the StreamGraph software package to visualize the trend of tweets across a period of time for all brands. In this study they analyzed sentiment polarity of more than 3500 social media tweets expressing attitudes towards sixteen global brands.

3. PROPOSED WORK

In the proposed method the Ngram technique is combined with the POS tagged data. Figure. 9 is the diagrammatic representation of the flowchart of the proposed method. POS tagger is a tool developed by Stanford University that can tag the parts of speech in a file. Essentially this tool takes as input a file containing movie reviews and gives an output in the form of annotated text. Every term in the annotated text contains a word and its associated parts of speech tag.

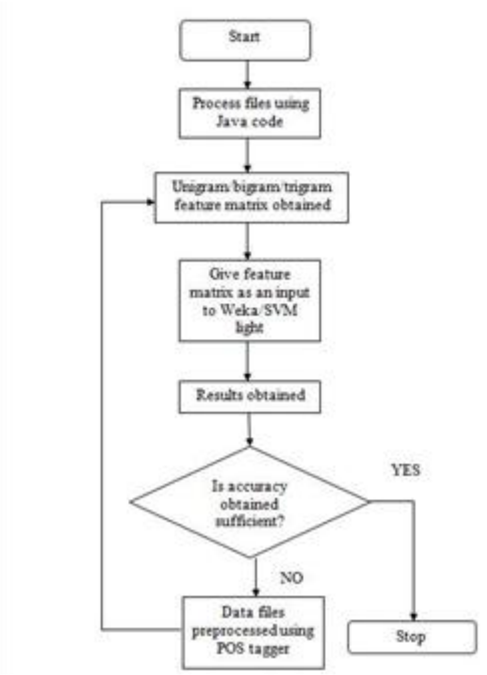

Start
Process files using Java code
Unigram/bigram/trigram feature matrix obtained
Give feature matrix as an input to Weka/SVM light
Results obtained
Is accuracy obtained sufficient?
YES
NO
Data files preprocessed using POS tagger
Stop

Fig:9 Flowchart

3.1 Creation of unigram feature matrix

The first experiment with the dataset was creating unigram feature matrix. The classifiers do not take plain text files as input; they require a feature matrix for training and testing. A Java code that creates a feature matrix from the text files containing movie reviews was written. The Java code calculates the term frequency of terms present in the text file that contains a movie review. Here term frequency is the number of times a particular term or word occurs in a file divided by the total number of words in the file. Here it is considered that each unique word occurring in the file is a unique feature and the term frequency of that particular word is the feature value. The Java code constructs a feature matrix with 2000 features.

Each entry in the feature matrix is the term frequency of each term occurring in the text files. The rows in the matrix specify the particular file containing the review whilst the columns specify the words present in each file. Unigrams are the single tokens that are extracted from the text file. If a sentence contains *n* distinct words then we would have *n* unigrams. This unigram feature matrix thus obtained was given to the classifiers and the result was obtained. Figure.10 shows how text is converted into unigrams.

A tenfold cross validation in Weka was performed for training and testing the classifier. The linear kernel in SVM lite was used for classification of the dataset.

3.2 Creation of bigram feature matrix

To improve the percentage of accuracy obtained a bigram feature matrix was created. Bigrams are two consecutive words extracted from a sentence. If a sentence contains *n* distinct terms, there would be n-2 bigrams formed from the sentence. The Java code calculates the term frequency for each unique bigram occurring in the text file containing the movie review. Another feature matrix is prepared where the features are the inverse document frequencies (idf) of each term occurring in the text file. This bigram feature matrix was given as an input to the classifiers and results were obtained. The results were obtained using a tenfold cross validation performed by the classification algorithms present in Weka, while the SVM light used a linear kernel for training and testing. Figure. 11 shows how text is converted into bigrams.

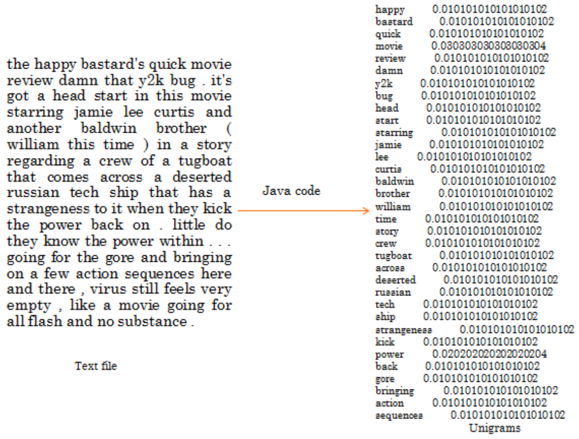

the happy bastard's quick movie review damn that y2k bug . it's got a head start in this movie starring jamie lee curtis and another baldwin brother (william this time) in a story regarding a crew of a tugboat that comes across a deserted russian tech ship that has a strangeness to it when they kick the power back on . little do they know the power within . . . going for the gore and bringing on a few action sequences here and there , virus still feels very empty , like a movie going for all flash and no substance .
Text file
Java code
happy 0.0101010101010101 02
bastard 0.0101010101010101 02
quick 0.0101010101010101 02
movie 0.0303030303030304
review 0.0101010101010101 02
damn 0.0101010101010101 02
y2k 0.0101010101010101 02
bug 0.0101010101010101 02
head 0.0101010101010101 02
start 0.0101010101010101 02
starring 0.0101010101010101 02
jamie 0.0101010101010101 02
lee 0.0101010101010101 02
curtis 0.0101010101010101 02
baldwin 0.0101010101010101 02
brother 0.0101010101010101 02
william 0.0101010101010101 02
time 0.0101010101010101 02
story 0.0101010101010101 02
crew 0.0101010101010101 02
tugboat 0.0101010101010101 02
across 0.0101010101010101 02
deserted 0.0101010101010101 02
russian 0.0101010101010101 02
tech 0.0101010101010101 02
ship 0.0101010101010101 02
strangeness 0.0101010101010101 02
kick 0.0101010101010101 02
power 0.0202020202020204
back 0.0101010101010101 02
gore 0.0101010101010101 02
bringing 0.0101010101010101 02
action 0.0101010101010101 02
sequences 0.0101010101010101 02
Unigrams

Fig.10 Conversion of text file into unigrams

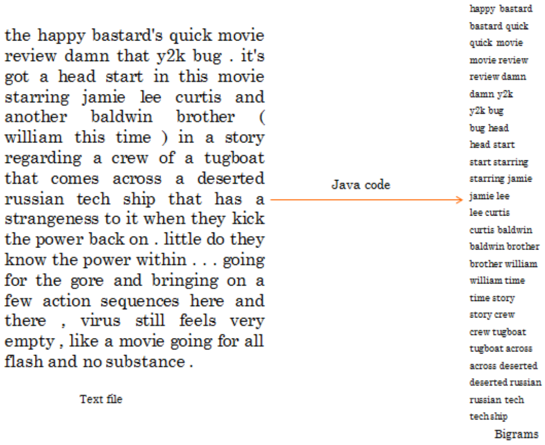

the happy bastard's quick movie review damn that y2k bug . it's got a head start in this movie starring jamie lee curtis and another baldwin brother (william this time) in a story regarding a crew of a tugboat that comes across a deserted russian tech ship that has a strangeness to it when they kick the power back on . little do they know the power within . . . going for the gore and bringing on a few action sequences here and there , virus still feels very empty , like a movie going for all flash and no substance .
Text file
Java code
happy bastard
bastard quick
quick movie
movie review
review damn
damn y2k
y2k bug
bug head
head start
start starring
starring jamie
jamie lee
lee curtis
curtis baldwin
baldwin brother
brother william
william time
time story
story crew
crew tugboat
tugboat across
across deserted
deserted russian
russian tech
tech ship
Bigrams

Fig. 11 Conversion of text file into bigrams

3.3 Creation of POS tagged bigram feature matrix

Finally the POS tagged output was used to modify the feature matrix and give more weightage to the specific terms in the bigram feature matrix. In this case the Java code constructs features that are bi-grams made up of an adjective and noun combination. Since the goal is to analyze the sentiments and classify them into two classes namely positive and negative we consider the adjective and noun combinations. Nouns are entities and adjectives are the qualifiers, giving more weightage to their combinations would bring out the positive or negative polarity of the sentences. There are adjectives that convey positive polarity; this would give the feature values of positive movie reviews a distinction as compared to the feature values of the adjectives conveying a negative sentiment. Thus the classifier would be trained to classify positive and negative opinions based on the features having more weights.

Here more weight is assigned to the bigrams containing noun-adjective combinations. Since adjectives are the parts of speech that qualify a noun, we give these terms more importance as compared to other bigrams. This modified feature matrix acts as an input for the classifiers and the final result is obtained. Figure. 12 shows how text is tagged using a POS tagger.

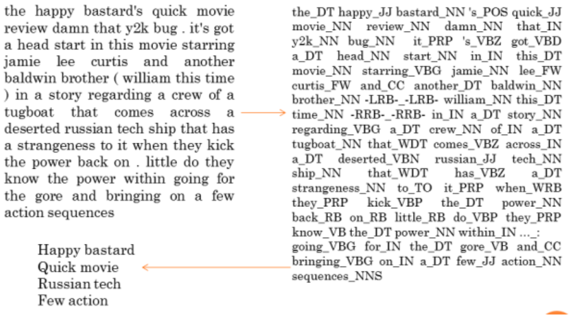

the happy bastard's quick movie review damn that y2k bug . it's got a head start in this movie starring jamie lee curtis and another baldwin brother (william this time) in a story regarding a crew of a tugboat that comes across a deserted russian tech ship that has a strangeness to it when they kick the power back on . little do they know the power within going for the gore and bringing on a few action sequences
the_DT happy_JJ bastard_NN 's_POS quick_JJ movie_NN review_NN damn_NN that_IN y2k_NN bug_NN it_PRP 's_VBZ got_VBD a_DT head_NN start_NN in_IN this_DT movie_NN starring_VBG jamie_NN lee_FW curtis_FW and_CC another_DT baldwin_NN brother_NN -LRB-_-LRB- william_NN this_DT time_NN -RRB-_-RRB- in_IN a_DT story_NN regarding_VBG a_DT crew_NN of_IN a_DT tugboat_NN that_WDT comes_VBZ across_IN a_DT deserted_VBN russian_JJ tech_NN ship_NN that_WDT has_VBZ a_DT strangeness_NN to_TO it_PRP when_WRB they_PRP kick_VBP the_DT power_NN back_RB on_RB little_RB do_VBP they_PRP know_VB the_DT power_NN within_IN ..._: going_VBG for_IN the_DT gore_VB and_CC bringing_VBG on_IN a_DT few_JJ action_NN sequences_NNS
Happy bastard
Quick movie
Russian tech
Few action

Fig.12 POS tagged output

Finally the efficiency of the intermediate experiments done to arrive at the final approach, are compared. The efficiency of the proposed approach is compared to the ones used in the previous research works.

4. RESULTS

Different classifiers that is; data mining and machine learning algorithms were used to obtain the results. Initially the unigram feature matrix was used to obtain the results. For unigrams it was observed that the best accuracy was given by rule based bagging method for tf-idf weights. Also it was seen that for unigrams, tf-idf weights gave a better accuracy in the categories: rules, meta and Bayes as compared to the term frequencies, whereas term frequencies gave a better accuracy in the categories: trees, lazy and function. Below are the results that were obtained from experiments on unigrams.

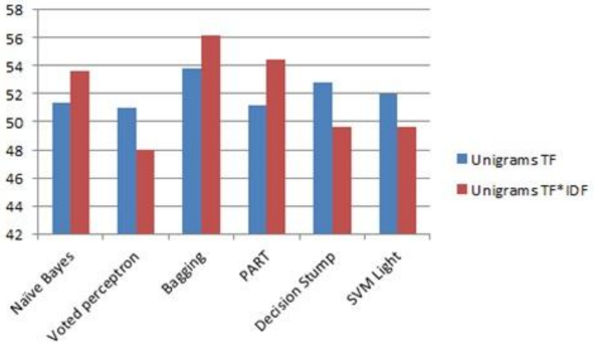


Fig. 13 Results for unigrams

For bigrams, it was observed that term frequencies give a better result as compared to tf-idf weights except for rule based and tree based classifiers. Figure. 14 shows the results that were obtained from the experiments on bigrams.

The best accuracy was obtained for POS tagged feature matrices. Here, the highest accuracy was obtained for SVM light using the term frequency matrix. For POS tagged terms Bayes, rules and tree based classifiers gave a better result for term frequencies. The percentage of accuracy obtained is 76.6 percent for SVM light. Figure. 15 shows the results obtained for POS tagged bigrams.

Three different methods were used to arrive at the final result. Classification of movie reviews was performed using unigram, bigram and POS tagged bigram. Below is Table 1 showing the comparison between the results of the different experiments that we performed on the dataset.

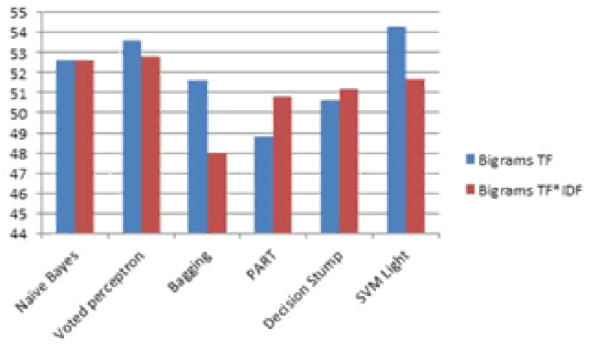


Fig. 14 Results for bigrams

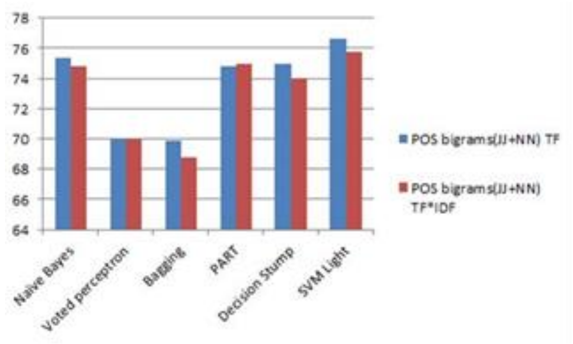
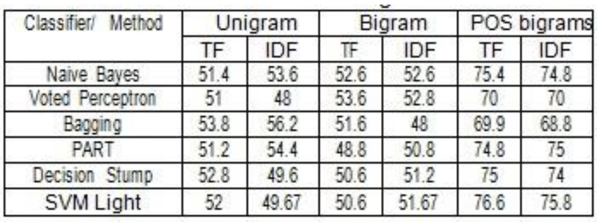


Fig. 15 Results for POS tagged bigrams

Table1 Comparison between unigrams, bigrams and POS bigrams



Classifier/ Method	Unigram		Bigram		POS bigrams	
	TF	IDF	TF	IDF	TF	IDF
Naive Bayes	51.4	53.6	52.6	52.6	75.4	74.8
Voted Perceptron	51	48	53.6	52.8	70	70
Bagging	53.8	56.2	51.6	48	69.9	68.8
PART	51.2	54.4	48.8	50.8	74.8	75
Decision Stump	52.8	49.6	50.6	51.2	75	74
SVM Light	52	49.67	50.6	51.67	76.6	75.8

An improved accuracy was achieved as compared to the approaches used earlier for sentiment analysis of movie reviews. Below is a Table. 2 showing the comparison of the results using POS tagged bigrams with the previous approaches.

Table 2. Comparison between existing techniques and our method

Method	Accuracy
Turney	65.83
Kim and Hovy	75.84
Bakliwal et al.	76.35
Das and Balabantaray	76.6

5. CONCLUSION

This paper contains the results of three experiments performed on a dataset of movie reviews. NLP and machine learning techniques have been used to obtain final results of the experiments. For SVM lite, a single kernel function is used to arrive at the results. In future this can be improved by using a larger dataset and by using multiple kernels for SVM lite. Also this work can be extended to cover other domains like product reviews, automobile reviews, hotel reviews etc.

6. REFERENCES

[1] Nitin Indurkhya, , Fred J. Damerau, “Sentiment analysis and subjectivity,” in In Handbook of natural language processing, Second Edition, 2009, pp. 627-630.

[2] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan, “Thumbs up? Sentiment classification using machine learning techniques,” in Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 7986, 2002.

[3] Peter Turney, “Thumbs up or thumbs down?,” in Semantic orientation applied to unsupervised classification of reviews, pages 417424, 2002.

[4] Kim, Soo-Min, and Eduard Hovy, “Determining the sentiment of opinions,” in Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics, 2004.

[5] Pang, Bo, and Lillian Lee, “A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts,” in Proceedings of the 42nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2004.

[6] Farah Benamara, Carmine Cesarano, Antonio Picariello, Diego Reforgiato, and V. S. Subrahmaniani, “Sentiment analysis: Adjectives and adverbs are better than adjectives alone,” in Proceedings of the International Conference on Weblogs and Social Media (ICWSM),2007.

[7] John Blitzer, Mark Dredze, and Fernando Pereira, “Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification.,” in Proceedings of the 45th Annual meeting of the Association of Computational Linguistics, pages 440-447,2007.

[8] Bakliwal, Akshat, et al., “Towards Enhanced Opinion Classification using NLP Techniques,” Sentiment Analysis where AI meets Psychology (SAAIP), 2011.

[9] Bespalov, Dmitriy. Et al., “Sentiment classification based on supervised latent n-gram analysis”, in Proceedings of the 20th ACM International conference on Information and Knowledge Management, ACM 2011.

[10] Wang, Hao et al., “A system for real time twitter sentiment analysis of 2012 US presidential election cycle” in Proceedings of the ACL 2012 System Demonstrations. Association for Computational Linguistics, 2012.

[11] Mostafa, Mohamed M. “More than words: Social networks text mining for consumer brand sentiments”, in Expert Systems with Applications, pages 4241-4251, 2013.