

# Towards Enhanced Opinion Classification using NLP Techniques

**Akshat Bakliwal**

SIEL, IIIT-Hyderabad  
akshat.bakliwal@research.iiit.ac.in

**Piyush Arora**

SIEL, IIIT-Hyderabad  
piyush.arora@research.iiit.ac.in

**Ankit Patil**

SIEL, IIIT-Hyderabad  
ankit.patil@research.iiit.ac.in

**Vasudeva Varma**

SIEL, IIIT-Hyderabad  
vv@iiit.ac.in

## Abstract

Sentiment mining and classification plays an important role in predicting what people think about products, places, etc. In this piece of work, using basic NLP Techniques like NGram, POS-Tagged NGram we classify movie and product reviews broadly into two polarities: Positive and Negative. We propose a model to address the problem of determining whether a review is positive or negative, we experiment and use several machine learning algorithms Naive Bayes (NB), Multi-Layer Perceptron (MLP), Support Vector Machine (SVM) to have a comparative study of the performance of the method we devised in this work. Along with this we also did negation handling and observed improvements in classification. The algorithm we proposed achieved an average accuracy of 78.32% on movie and 70.06% on multi-category dataset. In this paper we focus on the collective study of Ngram and POS tagged information available in the reviews .

## 1 Introduction

“What people think and feel” is the most important information for a business to promote and improve their product or for a production house to hit the blockbuster. Reviews are increasing with a rapid speed and are available over internet in natural language. This proves to be of utmost use for consumers and also for the manufacturers to improve the performance of their product. Sentiment analysis tries to classify reviews on the basis of their polarity either 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. The ratings and the reviews of the products helps the user to have a better overview of the product and make a choice based on overall rating of multiple reviews of the same product. In this paper, we propose a method to classify reviews as positive or negative. We devised a new scoring function and test on two different approaches which are

- Simple NGram (N=1/2/3) 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 (JJ) and Adverbs (RB) are used for scoring a review.

In another variant we used a combination of simple Ngram and POS-Tagged Ngram approaches. Based on the final score of a review it is classified as positive or negative. We also applied machine learning algorithms Naive Bayes(NB), Multi-Layer Perceptron (MLP) and Support Vector Machine (SVM) to study the performance of our method. The method was applied on two datasets movie review and product review.

In section 2, we describe the related work done in the past. Section 3, describes the algorithm proposed by us in this work. Section 4, describes tools, techniques and data used here. Section 5, focus on the experiments done and results of same. In section 6, small discussion over the results is done. Section 7, gives a conclusion of the present work.

## 2 Related Work

Identifying the sentiment polarity is a complex task, to address the problem of sentiment classi-

fication various methodologies have been applied earlier. Following are Unsupervised approaches.

1. Syntactic approach towards sentiment classification using Ngrams. This approach was used by Pang et al.(Pang et al., 2002) in their work.
2. Semantic approach using part of speech information. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews (Turney, 2002) and Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone (Benamara et al., 2007) used this approach for binary classification.
3. Extracting sentiment expressions using various NLP techniques. Sentiment Analysis: Capturing Favorability Using Natural Language Processing (Nasukawa and Yi , 2003) and Extracting Appraisal expressions (Bloom et al., 2007) used techniques like word sense disambiguation, chunking, n-gram and others to perform binary polarity classification.

Supervised approach uses machine learning supervised algorithms. Sentiment Classification for Chinese Reviews Using Machine Learning Methods Based on String Kernel (Zhang et al., 2008), Pang et al.(Pang et al., 2002), Twitter Sentiment Classification using Distant Supervision (Go et al., 2009) deduced some features to perform supervised machine learning.

Pang et al.(Pang et al., 2002) 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 three fold 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 (Turney, 2002) also worked on POS information. He used some tag patterns with a window of maximum three words (i.e) till trigrams.

In his experiments, he considered JJ, RB, NN, NNS 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.

Dave et al.(Dave et al., 2003) devised their own scoring function which was probability based. They performed some lexical substitutions to negation handling and used rainbow classifiers to decide the class of the review.

Our work is motivated from each of these works. Pang et al.(Pang et al., 2002) used POS information with unigram, we extended this work using POS information with bigrams and trigrams. Turney (Turney, 2002) also used POS<sup>1</sup> information with trigrams but he restricted trigram formation with some rules. He used PMI to evaluate the classification and here in this research we propose a new scoring function to classify. Dave et al.(Dave et al., 2003) devised some rules for negation handling and thus motivated us to work on negation handling.

### 3 Algorithm

To perform polarity classification we devised our own algorithm. This algorithm was applied on all our approaches. In our experiments we performed 5-fold cross-validation and we divided the pre-annotated data into two parts namely training set and testing set to check the correctness. After dividing the data we form trigrams, bigrams and unigrams on the training data and store them in individual n-gram dictionary. We create two separate models each for positive and negative polarity. For every testing review we create trigrams in the similar manner. Then we check if this trigram exists in our positive and negative trigram dictionary. If it exist then, we increase the count of trigram matched else we 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

<sup>1</sup><http://nlp.stanford.edu/software/tagger.shtml>

dictionary. Refer *Figure 1* for diagrammatic representation of algorithm.

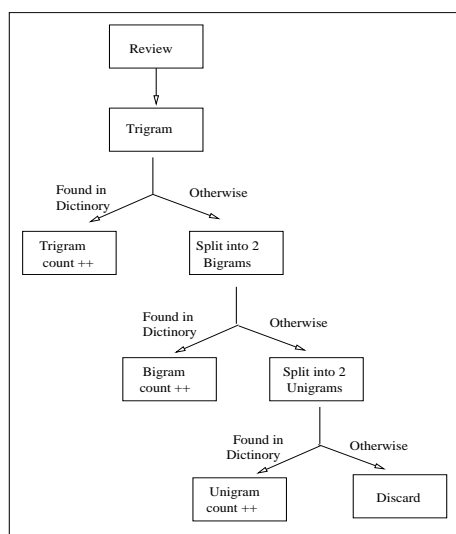


Figure 1: Algorithm Flow

We also propose a scoring function which gives priority to trigram matching followed by bigrams and unigrams.

$$\begin{aligned}
 \text{Score} = & x * \text{Count\_Tri} - \text{gram} + \\
 & y * \text{Count\_Bi} - \text{gram} + \\
 & z * \text{Count\_Uni} - \text{gram}
 \end{aligned}$$

here  $x = 7/11$ ,  $y = 3/11$ ,  $z = 1/11$ ,  $\text{Count\_N-gram} = \text{Number of N-grams matched (N = Uni/Bi/Tri)}$ . The values 7,3,1 are chosen to ensure that (1) score for matching a trigram  $>$  score for matching 2 bigrams. (2) score for matching a bigram  $>$  score for matching 2 unigrams. In the scoring function we have given the least possible integer value to unigram, bigram and trigram keeping the above constraints in mind. The rationale behind having these constraints while deciding the values of  $x$ ,  $y$ ,  $z$  was that higher n-gram carries more weight than a lower n-gram and also matching of a higher n-gram should be weighed more than matching of two lower n-grams. Then we have normalized these values on a scale of 0 to 1. So the final  $x$ ,  $y$ ,  $z$  parameters are  $x=7/11$ ,  $y=3/11$  and  $z=1/11$ .

## 4 Framework

This section describes various tools, techniques and data used by us in this work. We are using two different datasets in this work. One is Product Review dataset (*Refer Section 4.2.1*) which has reviews on multiple products belonging to different categories like apparels, books, software, etc.

This dataset is a multi category dataset in contrast to the other dataset which has only one category i.e. movies. Movie review dataset (*Refer Section 4.2.2*) contains reviews on various movies by critiques.

### 4.1 Tools and Algorithms

This section provides a brief details of the machine learning algorithms used in the experiments.

#### 4.1.1 Naive Bayes (NB)

Naive Bayes Classifier uses Bayes Theorem, which finds the probability of an event given the probability of another event that has already occurred. Naive Bayes classifier performs extremely well for problems which are linearly separable and even for problems which are non-linearly separable it performs reasonably well. We used the already implemented Naive Bayes implementation in Weka<sup>2</sup> toolkit.

#### 4.1.2 Multi-Layer Perceptron (MLP)

Multi Layer perceptron (MLP) is a feed-forward neural network with one or more layers between input and output layer. Feed-forward means that data flows in one direction from input to output layer (forward). We used the already implemented MLP in Weka toolkit.

#### 4.1.3 Support Vector Machine (SVM)

This classifier constructs N-dimensional hyper-plane which separates data into two categories. SVM models are closely related to a Neural Network. SVM takes the input data and for each input data row it predicts the class to which this input row belongs. SVM works for two class problems and is a non probabilistic binary linear classifier. We used libSVM<sup>3</sup> classifier which is available as a add on to Weka toolkit.

### 4.2 Datasets

This section provides a brief details of the datasets used by us in our experiments.

#### 4.2.1 Product Review Dataset

Multi-Domain Sentiment Dataset (Version 2.0)<sup>4</sup> (Blitzer et al., 2007) contains product reviews taken from Amazon.com belonging to different (total 25) categories like apparels, books, toys and

<sup>2</sup><http://www.cs.waikato.ac.nz/ml/weka/>

<sup>3</sup><http://weka.wikispaces.com/LibSVM>

<sup>4</sup><http://www.cs.jhu.edu/~mdredze/datasets/sentiment/>

games, videos, etc. We considered 4000 positive and 4000 negative reviews randomly sampled from 5 domains. Domains were chosen with utmost care so that they can represent non intersecting domains and 800 reviews of each polarity i.e. positive and negative are taken from each domain.

#### 4.2.2 Movie Review Dataset

Polarity Dataset (Version 2.0)<sup>5</sup> (Pang and Lee , 2004) contains 1000 positive and 1000 negative processed movie reviews on various movies. Reviews are pre-processed and divided into two categories positive and negative.

## 5 Experiments

We performed various experiments on the review data which were based on NLP techniques like n-gram, POS-Tagged n-grams, etc. We divided our work in two approaches.

### 5.1 Simple NGram Approach

While classifying the review the lexical information plays a very important role. The lower order n-grams i.e. unigrams and bigrams does not carry much information as compared to the higher order n-grams like trigrams or beyond. For example consider the phrase “not good product”, here unigrams formed are ‘not’, ‘good’ and ‘product’ but they does not carry sufficient information for polarity classification. When we move to bi-grams “not-good” and “good product”, “good-product” has a sentiment towards positive polarity and “not-good” is negating the positivity of good but the trigram “not good product” gives enough information to classify the trigram in negative class.

We experimented with different N-grams variation (unigram, bigram and trigrams) and its combinations (unigram + bigram and unigram + bigram + trigram ). The results (*Refer Table 1*) shows that the presence of trigrams with bigrams and unigrams has a favourable effect on classification of the reviews as positive and negative.

### 5.2 POS-Tagged NGram Approach

In this approach we used the part of speech information to deduce the opinion and subjective information in a given text. Adjective and Adverbs play an important role in deducing the subjective information since they reflect the qualitative judgment about a text. In this approach we create

<sup>5</sup><http://www.cs.cornell.edu/People/pabo/movie-review-data/>

	Movie Reviews	Product Reviews
Unigram only	64.1	42.91
Bigram only	76.15	69.62
Trigram only	76.1	71.37
(Uni + Bi) gram	77.15	72.94
(Uni + Bi + Tri) gram	<b>80.15</b>	<b>78.67</b>

Table 1: Results of Simple NGram

	Movie Reviews	Product Reviews
POS-(U + B + T)-JJ	75.00	50.425
POS-(U + B + T)-RB	65.50	36.76
POS-(U + B + T)-(JJ + RB)	<b>76.50</b>	<b>62.06</b>

Table 2: Results of POS-Tagged NGram. U = Unigram, B = Bigram, T = Trigram

trigrams, bigrams, unigrams of only those words whose part-of-speech tag is either Adjective (JJ) or Adverb (RB). For trigram we have \*-JJ/RB-\*, Bigrams \*-JJ/RB or JJ/RB-\* and unigrams are JJ/RB. here \* signifies any other pos-tag. Consider this review “This is a good product”, POS-tagged output of this review is “this\_DT is\_VBZ a\_DT good\_JJ product\_NN”. For this review we have 1 trigram “a-good-product”, 2 bigrams “a-good” “good-product” and 1 unigram “good”.

Similarly we find possible n-grams for RB Tag. After forming these NGrams we apply our algorithm and based on the score we get from both the positive and negative model we deduce the nature of the opinion. *Table 2* reports the accuracy of our scoring function on the two datasets after considering different variation of POS-tags such as only adjectives (JJ), only adverbs (RB) and both combined together.

In a variation to the above approach, we also incorporated negation handling and observed an increment in the overall performance. For negation handling our approach was: first we identified all the words with pos-tag JJ/RB. Then for negation handling we took a sliding window of 1-3 words in left from that word. If any of the words in this window were string ‘not’ then we modified the original word by appending a # sign in front of it. This # sign signified that the word was preceded by a negative word. Consider this review “This is not a good product”, POS-Tagged

	<b>Movie Reviews</b>	<b>Product Reviews</b>
POS-(U + B + T)-JJ	75.80	51.50
POS-(U + B + T)-RB	65.9	37.55
POS-(U + B + T)-(JJ + RB)	<b>77.35</b>	<b>62.75</b>

Table 3: Results of POS-Tagged NGram with Negation Handling. U = Unigram, B = Bigram, T = Trigram

output of this review is “this\_DT is\_VBZ not\_RB a\_DT good\_JJ product\_NN”. Now for this review if we make trigrams, bigrams and unigrams without negation handling, they will be 2 trigrams “is-not-a” “a-good-product”, 4 bigrams “is-not” “not-a” “a-good” “good-product” and 2 unigram “not” “good”.

None of these n-grams show the effect of not on good but if we do negation handling then the n-grams formed will be 2 trigrams “is-not-a” “a-#good-product”, 4 bigrams “is-not” “not-a” “a-#good” “#good-product” and 2 unigram “not” “#good”.

After negation handling n-grams formed clearly indicates that the information of a negative word “not” preceded by good is incorporated. *Table 3* reports the accuracy of our scoring function on the two datasets after applying negation handling.

To assert the performance of our scoring function, we formed a feature vector with features very closely similar to our scoring function. In our scoring function we considered the count of n-grams matched and the feature vector is also formed with the same information. Features are selected in a way that they only differ in terms of weighted parameters(x, y, z) from the scoring function. Our feature vector composed of 6 features + class which are calculated from the annotated data. Our features were  $\langle \mathbf{PUM, PBM, PTM, NUM, NBM, NTM, class} \rangle$  where PUM = Positive Unigram Matched, PBM = Positive Bigram Matched, PTM = Positive Trigram Matched, NUM = Negative Unigram Matched, NBM = Negative Bigram Matched, NTM = Negative Trigram Matched and class = Actual class of the review. We formed this feature vector for both the above mentioned approaches.

### 5.3 Feature Vector Approach

In the above two approaches (N-gram and POS tagged approach) we devised our own scoring function and calculated the polarity of an opinion but it might be the case that the function we used are biased, so in this approach we divided the dataset into training and testing set and extracted the features for the training set and formed feature vector for each of the opinion, we used machine learning algorithms for classification. We used WEKA toolkit for classification of the testing set (opinions). The feature vector was devised for both approaches.

For NGram and POS-tagged feature vector was the same as mentioned above. *Table 4* reports the accuracy of machine learning approach on Simple NGram and POS-Tagged NGram approaches.

We also combined Approach 1 (Simple NGram) and Approach 2 (POS-Tagged NGram) and the results were as shown in *Table 5*. Feature Vector for the combined training was  $\langle \mathbf{PUM, PBM, PTM, NUM, NBM, NTM, pt-PUM, pt-PBM, pt-PTM, pt-NUM, pt-NBM, pt-NTM, class} \rangle$  where where PUM = Positive Unigram Matched, PBM = Positive Bigram Matched, PTM = Positive Trigram Matched, NUM = Negative Unigram Matched, NBM = Negative Bigram Matched, NTM = Negative Trigram Matched, pt-PUM = POS-Tagged Positive Unigram Matched, pt-PBM = POS-Tagged Positive Bigram Matched, pt-PTM = POS-Tagged Positive Trigram Matched, pt-NUM = POS-Tagged Negative Unigram Matched, pt-NBM = POS-Tagged Negative Bigram Matched, pt-NTM = POS-Tagged Negative Trigram Matched and class = Actual class of the review

## 6 Result Analysis

In this section we compare the performance of our algorithm with the machine learning algorithm. Our algorithm reported accuracy well in consistency with machine learning algorithms. Among the various experiments done in approach 1 (Simple NGram) for movie review dataset, our algorithm reports maximum accuracy for (unigram + bigram + trigram) which is 80.15 and close equivalent to machine learning algorithm. SVM reports 81.15 and MLP reports 81.05 accuracy for (unigram + bigram + trigram) combination. For product review dataset also, results are closely related. Our algorithm reports accuracy of 78.76

	Movie Reviews			Product Reviews		
	NB	MLP	SVM	NB	MLP	SVM
NGram Feature	75.50	81.05	<b>81.15</b>	62.50	79.27	<b>79.40</b>
POS-Tagged Feature	72.35	<b>76.35</b>	75.45	68.81	<b>70.87</b>	67.88
POS-Tagged Feature with Negation Handling	72.80	<b>76.65</b>	75.00	68.83	<b>70.95</b>	67.95

Table 4: Results of Approach 1 and Approach 2 on Machine Learning Algorithms

	Movie Reviews			Product Reviews		
	NB	MLP	SVM	NB	MLP	SVM
Simple + POS-Tagged NGram Feature	78.05	<b>81.60</b>	78.45	69.25	<b>79.47</b>	78.86
Simple + POS-Tagged NGram with Negation Handling Feature	79.35	<b>81.60</b>	78.50	69.17	<b>79.39</b>	79.03

Table 5: Results of Approach 1 Approach 2 Combined on Machine Learning Algorithms

while SVM reports 79.4 and MLP reports 79.27. This shows that our algorithm performs as good as supervised learning approach and the selection of the parameters  $x, y, z$  in our algorithm are close to accurate.

For approach 2 (POS-Tagged NGram), we observed a similar adjacency between our algorithm and machine learning. For movie review dataset our algorithm performed best for (JJ + RB + Negation Handling) and accuracy attained was 77.35 which is higher than that achieved using SVM (75) and MLP (76.65). In case of product review dataset accuracy attained by our approach was 62.75 while the machine learning algorithms SVM (67.95) and MLP (70.95) dominated.

An observation we made while experimenting was that our model performs well when the reviews are domain specific (i.e. movie review) but when it comes to a larger or multiple domains (multi category product reviews) our performance drops down. Possible reason behind this could be that when we train on multiple categories together there may be cases that a specific category performs poorly and thus it pulls the over all performance down.

Main problem while dealing with sentiment analysis on reviews is that reviews span over multiple sentences. There are cases when a review contains multiple sentences and among them few sentences have opposite sentiment. For ex. "This movie was superb, good dialogs and action. The plot was awful". In this review the first sentence shows positive polarity and the second sentence show negative polarity. It may be the case that though the review was rated posi-

tive by the reviewer but the negative scored dominated and hence our system classified this as negative. This problem sometimes also occur within the sentence. Consider this review "This mobile phone has awesome features but the camera really sucks". In this sentence, the part before 'but' is positive and the part after but is negative. This review is neither positive nor negative and fails while classifying.

## 7 Conclusion

Based on these basic experiments which are simple to understand and perform one can get a approximate idea of the sentiments carried by reviews. We have presented simple techniques which are not restricted to review domain. With small simple modifications one can extend this work to various spheres like blogs, news (though we have not tested for the same and thus we make no claims). We obtained a general increment of 2-5% from the work done previously. This work will provide enough help to business industry to analyze what consumers think about their company and products.

## Acknowledgments

We would like to thank Manisha Verma and Karan Jindal for extending their help, support and guidance during this work. We also extend our heartiest thanks to Prasad Pingalli of SETU Softwares for his guidance.

## References

- Farah Benamara, Carmine Cesarano, Antonio Picariello, Diego Reforgiato, and V. S. Subrahmanian. Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2007. Short paper.
- John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *ACL*, pages 187–205, 2007.
- Kenneth Bloom, Navendu Garg, and Shlomo Argamon. Extracting appraisal expressions. In *Proceedings of Human Language Technologies/North American Association of Computational Linguists*, 2007.
- Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. pages 519–528, 2003.
- Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. pages 1–6, 2009.
- Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. pages 174–181, 1997.
- Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture, K-CAP '03*, pages 70–77, New York, NY, USA, 2003. ACM.
- 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 79–86, 2002.
- Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271–278, 2004.
- Peter Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. pages 417–424, 2002.
- Changli Zhang, Wanli Zuo, Tao Peng, and Fengling He. Sentiment classification for chinese reviews using machine learning methods based on string kernel. In *Proceedings of the 2008 Third International Conference on Convergence and Hybrid Information Technology - Volume 02*, pages 909–914, Washington, DC, USA, 2008. IEEE Computer Society.