SENTIMENT ANALYSIS OF MOVIE REVIEWS

Kuber Kaul
Masters Candidate
Columbia University
503-West,122ndStreet
kk2872@columbia.edu

Dragomir R. Radev
Department of Linguistics
University of Michigan
Ann Arbor, Michigan
radev@umich.edu

ABSTRACT

In this paper, a method for automatic sentiment analysis of movie reviews is proposed, implemented and evaluated. The focus of this study is completely on determining sentiment orientation (positive versus negative), the proposed method performs fine-grained analysis to determine the sentiment orientation of various aspects of a movie. Sentences in review documents contain independent clauses that express different sentiments toward different aspects of a movie. This can be somehow related the adjectives in the sentence. The method adopts a classifier approach of computing the sentiment of a clause from the prior sentiment scores assigned to individual words, taking into consideration the grammatical dependency structure of the clause. Negation is delicately handled. The output sentiment scores can be used to identify the most positive and negative clauses or sentences with respect to particular movie aspects. There have been 3 different classifiers and 2 separate datasets which have been experimented with.

Programming Languages

The Project has been implemented in python language and the following Language Constructs and Features have been used: –

Algorithm – The algorithm we use here is basically the machine learning algorithms used to classify the movie reviews which are Naïve Bayes, KNN algorithm and the

Design – The design of the code is as follows in the order:

- **a.** Cleaning of data
- **b.** Extraction of feature vector
- **c.** Classification of reviews using machine learning algorithms

Languages – The language used to implement the project is Python.

Theory – The theory of this project is loosely based on appraisal theory that is the idea that emotions are extracted from our evaluations (appraisals) of events that cause specific reactions in different people. Essentially, our appraisal of a situation causes an emotional, or affective, response that is going to be based on that appraisal. An example of this is going on a first date. If the date is perceived as positive, one might feel happiness, joy, giddiness, excitement, and/or anticipation, because they have appraised this event as one that could have positive long term effects, i.e. starting a new relationship, engagement, or even marriage. On the other hand, if the date is perceived negatively, then our emotions, as a result, might include dejection, sadness, emptiness, or fear. (Scherer et al., 2001) [1] Reasoning and understanding of one's emotional reaction becomes important for future appraisals as well. The important aspect of the appraisal theory is that it accounts for individual variances of emotional reactions to the same event

KEYWORDS

Sentiment analysis; SVM; Naïve bayes; Bag of words; KNN classification algorithm; Python; Unigrams/Bigrams/Trigrams; Feature extraction; Mutual information based selection; Frequency based feature extraction.

1. INTRODUCTION

The approaches which have been given in the paper were studied by me and implemented as part of the Project. The problem of building a classifier for Polarity detection of Movie Reviews involves two major steps. The first step is to reduce the document to a feature vector. The feature vector should embody characteristics of the document that lead to an intuitive suggestion towards the polarity. This would primarily involve identification of features in the language used in the document. For the purpose of our analysis, all documents have been reduced to the adjectives present in them. The primary assumption here is that the adjectives carry the major semantic weight of the entire document that points towards its polarity. Once the feature vectors have been constructed for every document, the next step is to build a classifier using an appropriate algorithm. Support Vector Machines (SVMs) with a linear kernel have been used to learn a classifier on the feature vectors. I have also tested the dataset with the Naïve Bayes Classification Algorithm. To train and test the classifier, a huge set of Positive and Negative reviews were made available to us. I also included another dataset to test the consistency of the both the approaches across different datasets. Both these approaches have been described in more detail in the subsequent sections. Sentiment analysis aims to uncover the attitude of the author on a particular topic from the written text. Other terms used to denote this research area include "opinion mining" and "subjectivity detection". It uses natural language processing and machine learning techniques to find statistical and/or linguistic patterns in the text that reveal attitudes. It has gained popularity in recent years due to its immediate applicability in business

environment, such as summarizing feedback from the product reviews, discovering collaborative recommendations, or assisting in election campaigns.

The focus of this project is the analysis of the sentiments in the movie reviews to detect its polarity.

We expect the short comment to express succinctly and directly author's opinion on certain topic. We focus on two important properties of text:

a. Subjectivity – whether the style of the sentence is subjective or objective;

b. Polarity – whether the author expresses positive or negative opinion.

We are interested in the following questions:

1. To what extent can we derive the polarity of the movie reviews accurately?

2. What are the important features that can be extracted from the raw text that have the greatest influence on the classification?

3. What machine learning techniques are suitable for this purpose? We compare in total three techniques of supervised and unsupervised learning.

4. Are the short length database sizes more easily classified accurately than longer database size? We compare our machine learning algorithms to short size corpora to the larger existing corpus

We describe the experiments and interpret the results.

2. Normal Background and Related Work

Sentiment classification of reviews has been the focus of recent research. It has been attempted in different domains such as movie reviews, product reviews, and customer feedback reviews. It has been used to classify tweets on the basis of polarity and hence deciding the stock market predictions. We can thus say that Sentiment classification has indeed become very important topic of research. (Pang et al., 2002; Turney and Littman, 2003; Pang and Lee, 2004; Beineke et al., 2004; Gamon, 2004). Much of the research until now has focused on training Machine Learning algorithms such as Support Vector Machines (SVMs) to classify reviews. Research has also been done on positive/negative term-counting methods and automatically determining if a term is positive or negative (Turney and Littman, 2002) the title (Helvetica 18-point bold), authors' names (Helvetica 12-point) and affiliations (Helvetica 10-point) run across the full width of the page – one column wide. We also recommend phone number (Helvetica 10-point) and e-mail address (Helvetica 12-point). See the top of this page for three addresses. If only one address is needed, center all address text. For two addresses, use two centered tabs, and so on. For more than three authors, you may have to improvise.[1]

[1]Kaggle is a platform for predictive modeling and analytics competitions on which companies and researchers post their data and statisticians and data miners from all over the world compete to produce the best models.

3. Methodology

Our method of sentiment analysis is based upon machine learning. We explain what sources of data we used in 3.1, how we selected features in 3.2, and how we performed classification in 3.3.

3.1

I have used two different data sets to train and test the machine learning algorithms with. The different sources of data are:

- **a.** Training dataset consisting of 6398 movie reviews which were already classified to train our learning algorithm and a test dataset consisting of 4265 movie reviews on which we trained our machine learning classification algorithms to get the accuracy for the algorithm. The testing dataset was produced and set as a challenge in Kaggle* and we tested the algorithm on the dataset in Kaggle thus producing the accuracy percentage. The link to the competition in Kaggle is http://inclass.kaggle.com/c/cs6998/leaderboard. The dataset can be downloaded through the following link http://inclass.kaggle.com/c/cs6998/data.
- **b.** The alternate dataset that was used testing and training our machine algorithm was a set of 1000 positive reviews and positive negative reviews which was provided to me through a tutorial. The dataset has been submitted with the project report. The machine learning algorithm has been tested on the dataset and results have been predicted and reported.

3.2

There have been numerous feature selections that have been tried and tested in this project. The features extraction methods are namely:

3.2.1 Unigrams/Bigrams/Trigrams:

In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sequence of text or speech. An n-gram could be any combination of letters. However, the items in question can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus.

An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram" (or, less commonly, a "digram"); size 3 is a "trigram". Larger sizes are sometimes referred to by the value of n, e.g., "four-gram", "five-gram", and so on.

3.2.2 Stop words Removal:

There are certain words which don't make a difference to the result and might only produce the wrong result in case of Naïve Bayes as they may appear a large number of times. These words are called stop words are generally removed from the dataset so as to just concentrate on words that matter and should weigh in heavily on predicting the polarity of the movie review. Hence we use a very common English stop word list provided on

http://www.ranks.nl/resources/stopwords.html to remove the stop word from the movie reviews.

3.2.3 Stemming of word to root:

Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. We performed stemming on all the words in the movie review after we proceeded with the stop word elimination so as all the words with the same root are considered as a single word and hence it would increase the importance of the word and give it much more weight in the final prediction. Examples of word stemming would be:

car, cars, car's, cars' $\Rightarrow$ car

am, are, is $\Rightarrow$ be

3.2.4 Removal of any word of two characters or less:

Typically any word which might be a letter or two letters long is not a heavy weight word i.e. something that might contribute much towards the polarity of a movie review and would be unrequired if not removed through stop words elimination. Hence we remove all such things that might be up to 2 characters long. The important thing to note here is that it would also remove the punctuation marks that might occur a lot of times in the movie reviews and hence create error. Therefore we have removed such words/punctuation from the movie reviews.

3.2.5 Mutual information based selection:

Formally, the mutual information of two discrete random variables X and Y can be defined as:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\,p(y)} \right),$$

Mutual Information Based Selection The performance of the classifier may also be improved by removing some of the less useful features. We use expected Mutual Information as a measurement of a feature's usefulness. We divide each dataset into training (60%), tuning (20%), and testing (20%) subsets. Features were extracted from the training set and ordered by their MI scores. Top N were chosen to represent the documents in the tuning set, with N varying from top few features to the size of the feature space. Finally, for each dataset an N was chosen to maximize performance, and the testing set was used to determine classifier performance at this cutoff. Figure 2 shows the performance of the classifiers at various cutoff points for the tuning sets. For all datasets, the performance drops off as the number of features approaches 100% (the number of features in full feature space is different for each dataset). This means that when sorted by MI, the bottom features hurt the performance of the classifier. Towards the top of the list, the performance differs between the relatively small Pang & Lee dataset and the others, which are larger by an order of magnitude. We noted the best cutoff point for each dataset and use the testing set to get the accuracy scores of 0.798 (Pang & Lee) at 76% cutoff, 0.911 (Jindal) at 1%, and 0.837 (Blitzer) at 3%.

3.2.6 Frequency based selection:

Frequency based selection is another feature extraction method which has been used in our project to extract the scores for every valid word based on the number of times it may appear in the total dataset.

Frequency-based feature selection, which is, selecting the terms that are most common in the class. Frequency can be either defined as document frequency (the number of documents in the class c that contain the term t) or as collection frequency (the number of tokens of t that occur in documents in c). Document frequency is more appropriate for the Bernoulli model, collection frequency for the multinomial model.

Frequency-based feature selection selects some frequent terms that have no specific information about the class, for example, the days of the week (Monday, Tuesday, ...), which are frequent across classes in newswire text. When many thousands of features are selected, then frequency-based feature selection often does well. Thus, if somewhat suboptimal accuracy is acceptable, then frequency-based feature selection can be a good alternative to more complex methods. However, Figure 1 is a case where frequency-based feature selection performs a lot worse than MI and should not be used.

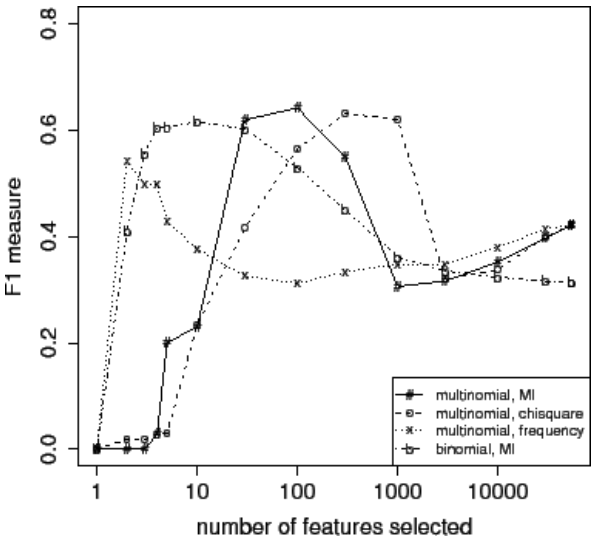

Figure 1: Effect of feature set size on accuracy

3.3 Classification techniques

Different classification algorithms were used and tweaked a little to fit the data set and produce the best accuracy possible. We were careful in not to overfit the dataset and hence output a poorer accuracy for any other dataset.

3.3.1 Bag of Words/Naïve-Bayes algorithm

The most basic model that I started working on was the bag of word model which I tweaked according to my needs, I started by extracting all the heavy-weight* words in a movie re-view. A universal hash of word root appearing in the positive reviews and the negative review were computed. Now any new review was tested against these using a number of distance functions. The standard dot product between the vectors was taken (i.e.: increments in a score would be the product of the number of times the adjective appears in the test review the number of times it appears in the positive hash).

Furthermore, a simple sum was done, i.e. the total number of word root that are positive minus the number of negative word root. Another tweak that I introduced was that a global hash of words that are generally shown positive and another hash of words that are generally considered negative were introduced and if any of the words in the database matched that of the positive word/negative word hash then there score was increased hence clearly increasing the weight. The idea behind this approach is that the words that appear in these hash lists greatly affect the polarity of the movie review and hence should be counted more. The cutoff for the minimum frequency for inclusion in the positive and negative universal hash was also tempered with, testing at 1, 5 and 10 respectively. Stemmers were used (stemmer used may be down- loaded from porter stemmer) though no perceivable change in the results was noticed. Furthermore, as suggested in, I negated all the adjectives in a sentence that appeared after a not or n't, though this too didn't bring about any appreciable change. We also experimented with all words with length > 2 were considered. The entire analysis was repeated for unigram, bigram and trigram. As mentioned by various papers on the same topic, I eliminated stop words and that did produce a hike in the accuracy as expected.

3.3.2 KNN classification algorithm

KNN was used for classification of movie-reviews. In this case, I took given some data points for training and also the other dataset for testing. My aim is to find the class label for the new point. The algorithm has different behavior based on k. Different k values were tried which produced different result. The most prominent ones which produced the best results were:

Case 1: k = 1 or Nearest Neighbor Rule

This is the simplest scenario. Let x be the point to be labeled. I found the point closest to x. Let it be y. Now nearest neighbor rule asks to assign the label of y to x. This is what nearest neighbor algorithm is, but this seems too simplistic and sometimes even counter intuitive. It will result in a huge error, but there is a catch. This reasoning holds only when the number of data points is not very large.

If the number of data points is very large, then there is a very high chance that label of x and y is same. An example of such nature is – Let's say you have a (potentially) biased coin. You toss it for 1 million time and you have got head 900,000 times. Then most likely your next call will be head. I am using a similar argument are.

Case 2 : k = K or k-Nearest Neighbor Rule

This is a straightforward extension of 1NN. Basically what I did is that I tried to find the k nearest neighbor and did a majority voting. Typically k is odd when the number of classes is 2. Example of that is let's say k = 5 and there are 3 instances of C1 and 2 instances of C2. In this case, KNN says that new point has to labeled as C1 as it forms the majority. We follow a similar argument when there are multiple classes. It is quite obvious that the accuracy might increase when you increase k but the computation cost also increases. In my experiments with k, I found out that the best accuracy was obtained with k=7 for the dataset that I used.

Table 1: Naïve-Bayes predictions on both the datasets.

Feature Extraction	**Classifier**	**Dataset Provided**	**Dataset-2**
Stop Word Elimination	NaïveBayes(Frequency based feature extraction)	75.559%	79.982%
Stop Word Elimination + Unigram	NaïveBayes(Frequency based feature extraction)	76.458%	81.012%
Stop Word Elimination + Bigram	NaïveBayes(Frequency based feature extraction)	75.359%	80.098%
StopWordElemination+Unigram+Positive/ Negative Word list.	NaïveBayes(Frequency based feature extraction)	77.788%	82.008%

Table 2: KNN - predictions on both the datasets (k=7)

Feature Extraction	Classifier	Dataset Provided	Dataset-2
Stop Word Elimination	KNN(Frequency based feature extraction)	65.099%	68.267%
Stop Word Elimination+ Unigram	KNN(Frequency based feature extraction)	65.599%	69.899%
Stop Word Elimination+ Bigram	KNN(Frequency based feature extraction)	65.012%	67.984%
StopWordElemination+Unigram+Positive/ Negative Word list.	KNN(Frequency based feature extraction)	66.023%	69.088%

3.3.3 Support Vector Machines

I used Support Vector Machine (SVMlight) V6.01 (Joachims 1998). It performs supervised learning by approximating a mapping

$$h: X \rightarrow Y$$

using labeled training examples (x1,y1), ..., (xn,yn). Unlike regular SVMs, however, which consider only univariate predictions like in classification and regression, SVM struct can predict complex objects y like trees, sequences, or sets. Examples of problems with complex outputs are natural language parsing, sequence alignment in protein homology detection, and markov models for part-of-speech tagging.

The SVM struct algorithm can also be used for linear-time training of binary and multi-class SVMs under the linear kernel

As explained in Dumais & Chen (2000) and Pang et al. (2002) given a category set, C = {+1, −1} and two pre-classified training sets, i.e., a positive sample set, Tr + Pn i=1(di, +1) and a negative sample set, T − r = Pn i=1(di, −1), the SVM finds a hyper plane that separates the two sets with maximum margin (or the largest possible distance from both sets).

At pre-processing step, each training sample is converted into a real vector, xi that consists of a set of significant features representing the associated document, di. Hence, Tr+ = Pn i=1(xi, +1) for the positive sample set and Tr− =Pn i=1(xi, −1) for the negative sample set.

Table 3: SVM classifier (using SVM light) on both the datasets

Feature Extraction	Classifier	Dataset Provided	Dataset -2
Stop Word Elimination	SVM (Frequency based feature extraction)	80.10%	81.56%
Stop Word Elimination + Unigram	SVM (Frequency based feature extraction)	80.98%	82.10%
Stop Word Elimination + Positive/Negative Word list.	SVM (Frequency based feature extraction)	81.34%	82.9%

4. Discussion

Since the recent New York Times piece on sentiment analysis, it seems everyone has an opinion on sentiment. The article uses sentiment analysis to refer to the industry, but sentiment analysis is better understood as just one of the types of analysis used in the field.

This industry has a history of picking up a new label almost every time someone new writes about it. Forrester Research has called it brand monitoring and listening platforms, depending on which year and analyst you ask. I picked social media analysis when I had to choose, but even that is more limited than the state of the art tools and services.

The results of this research can be a breakthrough in a lot of fields such as trade market sharing, social media analysis, voting trends etc.

Through my research and the study on the topic at hand , SVM holds the highest accuracy rendered. It provides enough accuracy to be practically used in an open data set with the confidence of getting a range of accuracy. Though this accuracy is pretty decent, but to really use it for more practical purposes we need to develop classifiers coupled with feature extraction methods and tweakers that hike up the accuracy to an almost perfect score or thereabouts. There have been a lot of research going on in the area and it can only lead to more positive results.

5. Conclusion

The results recorded were pretty interesting as I finished with Naïve Bayes showing higher percentages for accuracy of the two datasets used than KNN classification algorithm. SVM classification algorithm got the highest accuracy with both the datasets. Even though the Naïve Bayes is not so much of supervised learning than what KNN and SVM is and it takes into

account more features than a plain Naïve Bayes. The different feature extraction methods were also responsible for a hike in the accuracy rendered by the classifiers. The feature extraction tweaks were used to fit the dataset to obtain the best accuracy possible. The reason speculated for such a performance of KNN on our datasets is concluded to be:

a. Length of the data set: KNN performs better as the length of the dataset increases and here in our case with our dataset being around 4,500-5000 movie review long the performance of KNN dipped and the performance of Naïve Bayes actually increased a lot.

b. For shorter datasets Naïve Bayes can even outperform KNN classification algorithm or for that matter is competitive with SVM. Therefore we can say that length is directly proportional to the accuracy of the KNN classification algorithm whereas it's inversely proportional to Naïve Bayes algorithm.

c. We are only dealing with the reviews that we think are only positive or negative and not the ones that might be neutral but on close study I conclude that there is nothing as neutral review coz every movie review will finally tilt towards some polarity be that negative or positive.

Even though I did not test it officially for the project, **Learning Word Vectors for Sentiment Analysis by Andrew et al.** achieves about 88% accuracy on the same dataset we worked on and can be subject of interest for future work into the same topic.

One interesting to note was that using a lot of feature extraction methods can actually easily over fit the dataset and hence perform very badly against some other dataset. This was proved while experimenting as Naïve-Bayes with unigrams/stopword removal/stemming/different scoring pattern for positive and negative words/handling negative word I over fitted the dataset and ended up with an accuracy of 25% on the second dataset. The accuracy was regained to back to normalcy after I removed the different scoring pattern for positive and negative words based on a particular list of words.

6. Future Work

In the study, A. Turner noted that for movie reviews, the whole is not necessarily the sum of the parts". His suggestion is that often times, such as with the thwarted expectation narrative, we will have documents of one polarity which contain phrases distinctly of another polarity. Our scoring scheme, however, is very much based on a "sum of the parts" intuition. Our primary suggestion for future work is to explore alternate scoring schemes given our method of extracting sentiment items. One idea is to incorporate discourse structure into our classification. An example of this is subjectivity analysis, i.e. deter-mining when somebody is talking about the plot of a movie rather than their subjective opinion. Parts which are not subjective can interfere with a sum of the parts scoring approach, since the only parts we really want to consider are those which are subjective. A pertinent example would be a section of a movie review describing a horribly gruesome scene, but which nonetheless belongs to a positive review. Another idea is to incorporate sentiment into our classification scheme, which aims to model the global sentiment of a document as a trajectory of local sentiments. This can help identify thwarted narrative type reviews by helping the classier to understand more globally what the sentiment of the document is. Such cases are still to be handled properly through our scoring scheme and therefore needs to be thought about in our future work.

7. ACKNOWLEDGMENTS

My thanks to Professor Dragomir R. Radev for his guidance throughout the semester and for providing me with the relevant dataset and help with the paper.

8. References and citations

[1] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts, Stanford University. DOI = http://ai.stanford.edu/~amaas/papers/wvSent_acl2011.pdf

[2] Bo Pang and Lillian Lee, Department of Computer Science, Cornell University, Ithaca, NY 14853 USA. DOI= http://www.cs.cornell.edu/home/llee/papers/sentiment.pdf

[3] A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, Bo Pang and Lillian Lee, Department of Computer Science, Cornell University, Ithaca, NY 14853-7501. DOI= http://www.cs.cornell.edu/home/llee/papers/cutsent.pdf

[4] Sentiment Analysis: An Overview, Comprehensive Exam Paper, Yelena Mejova, Computer Science Department, University ofIowa. DOI= http://homepage.cs.uiowa.edu/~ymejova/publications/CompsYelenaMejova.pdf

[5] A Detailed Introduction to K-Nearest Neighbor (KNN) algorithm.DOI= http://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/

[6] Sentiment Analysis Tutorial. DOI= http://alias-i.com/lingpipe/demos/tutorial/sentiment/read-me.html

[7] Sentiment Analysis of Movie Review Comments, 6.863 Spring 2009 final project, Kuat Yessenov, Sasa Misailovi ˇ c´ . DOI= http://people.csail.mit.edu/kuat/courses/6.863/report.pdf

[8] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In Empirical Methods in Natural Language Processing [and Very Large Corpora].

[9] P.D. Turney. 2002. Thumbs 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.

[10] Vladimir Vapnik. 1998. Statistical Learning Theory. Wiley, Chichester, GB.

[11] Janyce Wiebe. 2000. Learning subjective adjectives from corpora. In Proc. 17th National Conference on Artificial Intelligence (AAAI-2000),Austin, Texas, July.

[12] J Wiebe. 2002. Instructions for annotating opinions in newspaper articles. Technical Report TR-02-101, University of Pittsburgh, Pittsburgh, PA.