International Journal of Computer Trends and Technology
ISSN: 2231 – 2803 / https://doi.org/10.14445/22312803/IJCTT-V69I3P115

Volume 69 Issue 3, 85-89, March, 2021
© 2021 Seventh Sense Research Group®

Original Article

Multi-Domain Sentiment Analysis

Sonali Aggarwal

Senior IEEE Member, Berkeley, CA, USA

Received Date: 16 February 2021
Revised Date: 30 March 2021
Accepted Date: 31 March 2021

Abstract - *I considered the problem of classifying amazon reviews by overall sentiment, i.e., positive or negative. Given a review that could be positive (4 or 5 stars) or negative (1 or 2 stars), the task is to accurately perform the binary classification. The review can be in any domain like books, electronics, DVDs, Kitchen, and others. However, in this project, I have limited myself to books, DVDs, and Kitchen. I investigated the performance of supervised machine learning methods like Naive Bayes, Support Vector machines(SVM), and Decision Trees for the problem of classification based on the overall sentiment of the reviewer. I also tested how well a classifier trained in one domain performs on the other domains.*

Keywords *- Amazon Reviews, Sentiment Analysis, Supervised Machine Learning.*

I. INTRODUCTION

The main goal of sentiment analysis is to determine the attitude of the writer towards some product domain. It aims to judge whether a review expresses a positive or a negative opinion. Sentiment analysis determines whether the user has a positive or a negative opinion of the item of interest. The main challenge in sentiment classification as compared to topical categorization. Topics can be identified by keywords. However, sentiment is more subtly expressed in text. Also, there might be sentences having a negative connotation, without any negative sentiment word (e.g., How could anyone use this product?). These factors make sentiment analysis a very interesting and challenging field of study.

II. THE PROBLEM AND METHODOLOGY

A. Description of the DataSet

I used the Multi-Domain Sentiment Dataset provided by Blitzer et al. [1]. It contains product reviews obtained from Amazon.com from many product types (domains such as books, DVDs, and kitchen appliances). Each review contains star ratings (1 to 5 stars), a reviewer name and location, a product name, a review title and date, and the review text. Reviews with a rating of more than 3 were labeled positive, those with a rating less than 3 were labeled negative, and the rest were discarded because they are ambiguous in nature. The data-set is fairly balanced, containing a proportionate amount of positive and negative sentiment reviews.

B. Overview of the Methodology

I tested various supervised learning approaches viz. Naive Bayes, Decision trees, and SVMs. In addition, I experimented with different features for these classification algorithms - unigrams, bigrams (two words from a document in sequential order) with and without stemming. I also experimented with one unsupervised approach, which is explained in the methodology section.

C. Primary Results

SVMs were found to have worked the best, with an average accuracy of 83%. Naive Bayes followed closely with an accuracy of 82%. Decision Trees gave us an accuracy of around 75%. This can be attributed to the fact that decision trees overfit the data since I have a very scarce data matrix.

III. PRIOR WORK

Bo Pang [3] explores supervised machine learning techniques to do sentiment classification of movie reviews. This work demonstrated that Naive Bayes-based text categorization performs well despite being a simplistic model and the fact that its conditional independence assumption does not work in real-world scenarios. The authors also explore MaxEnt, which works better when conditional independence assumptions are not met. Support vector machines (SVMs), which are large margin classifiers, also show good performance at traditional text categorization, generally outperforming probabilistic classification methods. The authors experiment with different features for these classification algorithms - unigrams, bigrams, POS tagging with unigrams, adjectives alone. They got the best accuracy with using unigrams as features on SVMs, though they did not see a significant difference between SVMs and Naïve Bayesian methods.

Turney [4] uses unsupervised learning approaches to do sentiment classification of movie reviews. They predict the classification of a review by the average semantic orientation of the phrases in the review that contain adjectives or adverbs. The first step of their algorithm is to use a part-of-speech tagger to identify phrases in the input text that contain adjectives or adverbs. The second step is to estimate the semantic orientation of each extracted phrase. The Semantic Orientation (SO) of a phrase, phrase, is calculated by PMI (Pointwise Mutual

Information), which is the mutual information between the given phrase and the word “excellent” minus the mutual information between the given phrase and the word “poor.” The authors deploy PMI-IR estimates to calculate PMI by issuing queries to a search engine (hence the IR in PMI-IR) and noting the number of hits (matching documents). The third step is to assign the given review to a class, recommended or not recommended. The algorithm achieves an average accuracy of 74%, ranging from 84% for automobile reviews to 66% for movie reviews. The main advantage of unsupervised learning is that no training data is required for classification.

Chaovalit and Zhou[5] also explore movie review mining. This paper compares and contrasts different machine learning methods to classify product reviews. The authors mention the challenges of movie review mining since factual information is always mixed with real-life review data, and ironic words are used in writing movie reviews. The Supervised learning (classification) approach tends to be more accurate because each of the classifiers is trained on a collection of representative data known as corpus. In contrast, using the semantic orientation approach (unsupervised learning) is not as accurate because it does not require prior training in order to mine the data. However, the supervised learning method has its drawbacks due to the dependency on corpus data. It needs retraining if it is to be applied elsewhere. They used 3-fold cross-validation for their supervised learning methods and derived an average accuracy of 85.54% when tested on the test dataset. The paper deploys Turney's approach to doing classification by unsupervised learning to achieve an accuracy of 77%.

These three papers compare different methods of doing sentiment analysis on movie reviews. I am using a different data-set -- Product Review Dataset on Amazon Website. The dataset I use is highly domain-specific, so I tested how these supervised and unsupervised approaches work across all the aforementioned domains. I learned how a model trained on one domain performs on the other domain.

IV. MY APPROACH

First, I created the input matrix using the methodology explained in section IV.A. After creating the input matrix, I applied the classification algorithms. Classification algorithm predicts the label for a given input sentence. There are two main approaches for classification: supervised and unsupervised. In supervised classification, the classifier is trained on labeled examples that are similar to the test examples. On the other hand, unsupervised learning techniques assign labels based only on the semantic orientation of the word it contains. I considered three supervised algorithms – Naive Bayes, Decision Trees, Support Vector Machines and one unsupervised classification approach

A. Feature Selection and Extraction

The XML input files are parsed to create an input file that has the review and the rating on each line. Each review was pre-processed to remove stop words (words which frequently occur such as a, and, the, etc.). The words of the review are stemmed from using Porter’s Stemming algorithm. I also added bigrams to the review. I used a bag-of-words model. Bag-of-words is a model that takes individual words (unigrams and bigrams) in a sentence as features, assuming their conditional independence. The text is represented as an unordered collection of these features. After pre-processing the review, I created an input matrix where each row is a review, each column is a feature, and the matrix value indicates how many times the feature occurs in the review. I obtained a feature vector of 35000-36000 words, and the number of training reviews I took is 2000. I had a test corpus for each domain containing 4500-6000 reviews.

B. Naive Bayes

I started with a Naive Bayes approach to model the reviews I have. In this approach, I first build a dictionary of all the words in the training and testing data. A review is represented as a feature vector of length V, where V is the dictionary size. I use X to denote the feature vector for a document and Y to denote the class label. I use the label 1 to denote positive reviews and 0 to denote negative reviews. For a review with feature vector X, Xi is the frequency count of the occurrence of the ith dictionary word in the review. The Naive Bayes assumption is that the Xi’s are conditionally independent given the review label Y. So in order to predict the label for a new document, I used the following equation -

$$\begin{aligned} p(y=1|x) &= \frac{p(x|y=1)p(y=1)}{p(x)} \\ &= \frac{\left(\prod_{i=1}^{n} p(x_i|y=1)\right) p(y=1)}{\left(\prod_{i=1}^{n} p(x_i|y=1)\right) p(y=1) + \left(\prod_{i=1}^{n} p(x_i|y=0)\right) p(y=0)}, \end{aligned}$$

and then picked the class with the highest posterior probability. I calculated all the quantities in the above equation using the training data.

Here P(Y = 1) is the prior probability of the review having a positive label. It is simply the fraction of reviews with a positive review. If ‘m’ is the total number of training documents and Yi denotes the label of the ith review, I can calculate the prior probability using this equation (Here {1} is the identity function which returns 1 if the argument evaluates to TRUE) –

$$p(y) = \frac{\sum_{i=1}^{m} 1\{y^i = 1\}}{m}$$

Here P(Xi | Y = 1) is the conditional probability of the ith dictionary word given a positive label of the document. It is basically the fraction of times the word appears in all positive reviews divided by the length of all positive reviews. The data is very sparse, and I may encounter many new words in the test data set, and their prior conditional probabilities would be zero, and if I were to use these, then I would not be able to make a prediction(I would get a 0/0 form). To solve this problem, I estimated the probabilities of unseen words using Laplace smoothing. Here I used Xji to denote the ith component of the feature vector for document j. nj is the length of the jth document, and V is the size of the vocabulary. I used the following formula to calculate the prior conditional probabilities -

$$p(x_i|y=1) = \frac{\sum_{j=1}^{m} 1\{y^j = 1\}x_i^j + 1}{\sum_{j=1}^{m} 1\{y^j = 1\}n_j + |V|}$$

Similarly, I calculated the parameters for negative reviews as well. I performed the experiment on three domains – Books, DVDs, and Kitchen. I built a separate model for each of the three domains and performed testing on each of them.

C. Decision Trees

Decision trees create a flowchart-based classifier. At each level, they use decision branches, a simple classifier that checks for the presence of a single feature. The label is assigned to the sentence at the leaf nodes of the tree. I used the ‘classregtree’ function in MATLAB Statistical toolbox to do the classification. Decision Trees are not as effective as Naive Bayesian methods for sentiment classification. The decrease in performance is due to the fact that a fairly large tree is needed to handle all of the feature attributes that are present in the datasets. Due to the inherent size of the tree, an unclassified testing instance has to traverse through many prediction nodes until it reaches a leaf node. The longer the path an instance has to travel, the higher the likelihood that an incorrect prediction will be made, thereby decreasing the classification performance on this task.

To avoid overfitting the training data-set by decision trees, I tried pruning the tree to reduce the number of levels. Pruning reduces the complexity of the classifier and removes sections that may be noisy. This improved the accuracy of results by 1 to 2%.

D. Support Vector Machines (SVM)

Support vector machines (SVMs) belong to supervised learning methods. They analyze data and recognize patterns by finding a hyperplane in an N-Dimensional space. SVM a non-probabilistic binary linear classifier. It predicts for each input the probability of belonging to each of the different classes.SVM training algorithm builds a model that predicts whether a new example falls into one category or the other.

Since the problem of Sentiment Analysis in Product Reviews can have a very large set of features (codewords), I decided to use SVM as they work even if the data matrix is sparse and have the ability to handle large feature vectors. For the purposes of binary classification, I used Thorsten Joachims’ SVM-light package Version 6.02 [9]. SVM-*light* is an implementation of Vapnik's Support Vector Machine [6], and the optimization algorithms used in SVM-*light* are described in [7] and [8]. The algorithm has scalable memory requirements and can handle problems with many thousands of support vectors efficiently. Also various Kernel options as linear, polynomial *(s a*b+c)^d*, radial basis function *exp(-gamma ||a-b||^2)* and sigmoid *tanh(s a*b + c)* Ire explored in this method.

E. Unsupervised NLP Approaches

I tried a simple unsupervised approach for sentiment classification in which review is predicted by the average semantic orientation of the phrases in the review that contain adjectives or adverbs. Even though the supervised learning approaches can be superior and more relevant to the classification task at hand, however, a large corpus of tagged training data must be collected and annotated, and this can be a challenging and expensive task.

A phrase’s semantic orientation depends on its associations; for example, the phrase “lovely fit” has positive associations, and the phrase “noisy atmosphere” has negative associations. The average semantic orientation of its phrases is then computed to predict if the review is positive or negative. I used General Inquirer (Stone et al., 1966) [2] wordlists to obtain the semantic orientation of words. Data in the General Inquirer set is maintained by Harvard University and consists of several hand-annotated dictionaries which attempt to place specific words into various descriptive categories. A wide range of adjective categories exists, including “positive," negative," pain," pleasure." Each word may belong to more than one category, and more than one instance of a word may exist depending on its contextual use. I use the categories of Diminishers, Intensifiers, Negate, Negative and Positive words. Negation words are used to reverse the polarity of the subsequent negative or positive word. Similarly, intensifiers and diminishers Are used to boost/subdue the polarity of the subsequent negative or positive term

V. RESULTS

A. Naive Bayes Classifier Results

Using Bigrams without Stemming gives the best accuracy. If I use stemming, the accuracy goes down by about 3 to 4%, but the advantage is that the dictionary size reduces by a factor of about one-third, which makes the processing much faster. Using only unigrams reduces the accuracy by about 2.5%, but again the advantage is that the dictionary size reduces by a huge factor of about 8 times which is a big saving on time and space requirements.

For Books, the words most indicative of positive reviews are – “beautifully,” “enjoyed this,” and “easy,” “loved this.” Words most indicative of negative reviews are – “waste,” “disappointing,” “dull,” “not worth.” For DVDs, the words most indicative of positive reviews are – “best movies,” “great story,” “highly recommended,” and the words most indicative of negative reviews are – “not worth,” “waste,” “unfunny,” “redeeming,” “atrocious,” “blah.” For Kitchen, the words most indicative of positive reviews are – “so easy,” “perfect,” “love this,” “a must,” “efficient,” “must-have,” “awesome,” “excellent product,” and words most indicative of negative reviews are – “waste,” “horrible,” “terrible,” “disappointment,” “not fit,” “junk,” “returning.”

From this, I can see that the reviews of Books and DVD's are domain dependant to some extent (Spiderman, Chaplin, Cinderella are considered good movies and Anderson a good author), so that's a reason why they have more accuracy compared to Kitchen, where the terms are very generic and can be applied to any domain.

Table 1. Results

Dataset	Accuracy with using Naive Bayes Classifier		
	Unigrams	Bigrams with Stemming	Bigrams with no Stemming
Books	80.4%	79.32%	82.68%
DVDs	80.48%	79.92%	82.77%
Kitchen	85.35%	83.72%	87.94%

B. SVM Results

The table below shows the overall accuracy for each of the positive and negative reviews when the features considered are unigrams, stemmed bigrams, and bigrams without stemming. Out of the three types of features extracted, bigrams without stemming gave the maximum overall accuracy in SVM for all the 3 domains considered.

Table 2. Results

Dataset	Accuracy with using SVM Classifier		
	Unigrams	Bigrams with Stemming	Bigrams with no Stemming
Books	80.51%	81.12%	86.59%
DVDs	80.81%	80.09%	83.67%
Kitchen	82.01%	81.79%	87.45%

C. Decision Tree Results

Decision trees have the worst accuracy suggesting that they do not handle data sets having too many features very well. The table below shows the overall accuracy, precision, and recall for both positive and negative reviews. The domain of books and dvds have a worse accuracy than that of Kitchen. This is expected since classifying books and movies is tougher since the sentiment of the story often interferes with the sentiment of the review. Also, pruning improves the result slightly.

3.Table Results

Dataset	Results with using Decision Trees		
	Precision	Recall	Overall Accuracy
Books	68.51%	67.5%	69.29%
DVDs	69.81%	73.40%	70.67%
Kitchen	75.01%	73.19%	74.45%

D. Unsupervised Classification Methods Results

The unsupervised approach explained earlier gave an average accuracy of only 70% on all domains. From the results, it is clear that even the worst of supervised learning approaches are better than the lexical approach. This method suffers from a huge drawback that only those words which are present in the General Inquirer Lexicon are given scores. Due to this, there is an upper bound on how well dictionary-based approaches can perform. Even though the supervised learning approach is superior, it requires a large corpus of tagged training data, collecting which is an expensive and challenging task.

VI. CONCLUSION

Table 3. Results

Dataset	Accuracy Values with Different Classifiers			
	Naive Bayes	SVM	Decision Trees	Unsupervised
Books	82.68 %	86.59 %	69.29%	67.59%
DVDs	82.77 %	83.67 %	70.67%	68.67%
Kitchen	87.94 %	87.45 %	74.45%	73.45%

I got the best accuracy with SVMs, using unigrams with bigrams, like features , without any stemming.
I got the best results with a linear kernel in SVMs, suggesting that the sentiment classification problem is linearly separable. Naive Bayes closely follows SVMs inaccuracy. Decision trees and unsupervised learning have similar performances. The impact of the stem is not as strong, which is expected since it has much more impact as a dimensionality reduction method. Adding bigram features considerably improves accuracy since I am using more information as features. Also, when a dataset trained on one domain is tested on another domain, the accuracy reduces considerably, which is expected since the training data is no longer relevant on a different domain.

VII. FUTURE WORK

I would like to experiment with classifying reviews according to a four or five-star rating, as opposed to a simple binary classification scheme. To further improve performance, I intend to use dimensionality reduction methods. The input matrix is extremely sparse as a review contains very few words, and most of the other feature words are 0. A proper dimensionality reduction like the Principal Component Analysis, Singular Value Decomposition can help us extract the most useful features in our data-set. I would also like to explore the combined use of multiple classifiers to predict sentiment by using ensemble methods. Ensemble methods aim at improving classification accuracy by aggregating the predictions from multiple classifiers, for example, by averaging results of classifiers that make errors independently of each other. This could lead to a significant improvement in results.

REFERENCES

[1] John Blitzer, Mark Dredze, Fernando Pereira. Biographies, Bollywood, Boom-boxes, and Blenders: Domain Adaptation for Sentiment Classification. Association of Computational Linguistics (ACL), (2007)

[2] STONE, P. J., DUNPHY, D. C., SMITH, M. S., OGILVIE, D. M., and associates. The General Inquirer: A Computer Approach to Content Analysis, The MIT Press, (1966).

[3] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan., Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of EMNLP, (2002). Introduced polarity dataset v0.9.

[4] Peter D., Turney: Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, ACL (2002)

[5] Pimwadee Chaovalit and Lina Zhou., Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches. Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS), (2005).

[6] Vladimir N. Vapnik., The Nature of Statistical Learning Theory. Springer, (1995).

[7] Thorsten Joachims., Learning to Classify Text Using Support Vector Machines. Dissertation, KluIr, (2002).

[8] T. Joachims, 11 in Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT Press, (1999).

[9] Thorsten Joachims' SVM-light package. Version 6.02. Date 14.08.2008 http://svmlight.joachims.org/

[10] Afreen Jha, N. Satya Deepthi, G.Suryakanth, G. Surya Kala Eswari., Text Sentiment Analysis Using Naive Bayes Classifier, IJCTT, (2020).

[11] Sai Kiran Chintalapudi, Harshavardhan Metla, Keerthi Shrikar , Nalluri Rahul., Opinion Mining and Sentiment Analysis for Amazon Product Reviews using Lexicon and Rule-Based Approach and Testing on Machine Learning Algorithms, IJCTT, (2018).

[12] Afreen Jaha, N.Satya Deepthi, G.Suryakanth, G. Surya Kala Eswari, Text Sentiment Analysis Using Naïve Baye's Classifier, IJCTT, (2020).