Review Mining and Intelligent Predictions

AshutoshBhargave
Indiana University
Bloomington
ashubhar@indiana.edu

Sohil Jain
Indiana University
Bloomington
sohjain@iu.edu

Abstract

The growth of social media over the last decade has revolutionized the way individuals interact and industries conduct business. Abstract Users of the online shopping site Amazon are encouraged to post reviews of the products that they purchase. The number of reviews for different products varies, but the reviews provide accessible and plentiful data for relatively easy analysis for a range of applications. This paper seeks to apply and extend the current work in the field of natural language processing and sentiment analysis to data retrieved from Amazon and apply predictions on twitter tweets extracted from same products to determine public opinion about the product. Naive Bayes and Support classifiers are used to tag a given review as positive or negative. The number of stars a user gives a product is used as training data to perform supervised machine learning. Our dataset contains 6,000 product review from three different products. Top selling and reviewed cellphones on the site are the primary focus of the experiments, but useful features of them that aid in accurate classification are compared to those most useful in classification of other products. The features, such as bag-of-words and bigrams, are compared to one another in their effectiveness in correctly tagging reviews. Errors in classification and general difficulties regarding the selection of features are analyzed and discussed. After training our classifier on Amazon, we have read twitter tweets about the same product and make predictions about its different characteristics/features. This can be used by the companies to analyze their products.

1 Introduction

The marketplace for all the consumer products has moved to the Internet. Because of the shift of users to the ecommerce world, the shopping experience has changed a lot. The concept of mouth publicity is decreasing day by day. Prior to Amazon and still today certain parts of businesses relies on the surveys to know about the user sentiments about the product. Most of the times the efforts carried out for conducting the survey has large amount of monetary issues associated with it.

Amazon has made product reviews and rating open to all the consumers. So, today anybody can go and post on Amazon about their experience of using a particular product. Anybody sitting at different places can make use of these reviews and take decision about buying a certain product. Today not just Amazon but people make use of social networking sites to post there review about the product. The most happening out of these is twitter. But how can these be useful to the companies or manufacturers of the product? With the increase in user-generated content, efforts have been made to understand the information in the correct context, and develop methods to determine the intent of the author. This has mostly been limited to the overall rating of the product till date. What about the particular feature of the product? Understanding what online users think of its product feature can help a company develop a particular feature and improve upon the ones which they lack at. With the help of this the company can market its product as well as mange its online reputation. This could also help the company to find out where it stands in comparison with their competitors on certain product feature. For example, X mobile company can find out that compared to products of similar range by Z Company, we have better camera but we lack in battery life so we should concentrate more on that.

The intent of this paper is to study this large problem of knowing the positive and negative attitude of users towards particular product features. Sentiment analysis attempts to determine which features of text are indicative of its context and build systems to take advantage of these fea-

tures. A lot of work involving sentiment analysis in review or opinion mining has been done relatively recently. With the use of different machine learning systems Pang and Lee (2002) classified a large number of movie reviews. Amazon employs a 1-to-5 scale for all products, regardless of their category, and it becomes challenging to determine the advantages and disadvantages to different parts of a product. That is what we are trying to focus on in our study.

2 Methodology

There are large amount of reviews about products on Amazon which facilitated us to do a supervised machine learning. We chose to analyze a few popular products namely *Iphone6,BLU Advanced*and *Asus Zenfone* because they had most number of reviews on Amazon and our machine could be trained properly.

In this project, we study the problem of generating product feature-based summaries of customer reviews of products sold online. Product features mean characteristics (or attributes) and functions of product like camera, battery, size, price, weight etc. Given a set of customer reviews on Amazon about a particular product, the task involves six subtasks:

(1) Identifying and storing the reviews and ratings of the product that customers have expressed their opinions on Amazon

(2) Identifying review sentences that give positive or negative opinions

(3) Training a classifier to classify the reviews

(4) Gathering the tweets about a particular product.

(5) Identifying features of the particular product that customers have expressed their opinions on (called as product features)

(6) Identifying tweets that give positive or negative opinion and predict the positivity with the help of the classifier we developed on Amazon reviews and provide summary of the same.

2.1 Data

Amazon: The reviews data was gathered from Amazon.com website. Amazon does not have an API like Facebook or Twitter to download the data. But, on Amazon there are links for the reviews for every product. We found out different patterns for URL representation and traversed the site on same. We also had an option of using the Stanford's Snap Amazon dataset. But, since the dataset is quite old and limited to Mobile reviews. We wanted the latest reviews specific to Mobiles from Amazon. We wrote a web crawler for this purpose. The web crawler was written in Java. We used JSOUP for the same. The code searches following things from the product

Review title, Review Text, Product review rating, Review Author, Review helpfulness.

Out of these we have focused on rating, title and text. After getting 10 reviews per page we iterate over X number of pages, Where,

$$X = \frac{\text{Total Number of reviews}}{10}$$

Extracting Amazon reviews: After gathering the reviews for a given product, the reviews are saved in a Tab Separated File (.tsv). Though we gathered all information related to reviews we focused only on review ratings, review text and review title for building our classifier. The Amazon website only allows rating from 1 to 5. For classification we distinguished as 0 and 1, as

Product	Reviews	Tweets	English Tweets
Iphone6	1328	12000	4000
Blu Advance	3786	-	-
Zenfone	2200	26000	15000
Note5	-	6000	2500

negative and positive respectively, which is explained in detail in experimentation part.

Twitter: We will use Twitter tweets as data to make our predictions after training our classifier on Amazon reviews. We gathered twitter tweets with the help of Tweepy[1] Rest API. The tweets were downloaded in the format of a JSON file. We imported the json data into MongoDB. After that we wrote a parser which retrieves all the data from MongoDB and stores it into a Tab separated file (.tsv). This saved lot of time from parsing complicated json data we could label the data properly in tsv files. From all the twitter data we gathered we focused on following parameters,

Tweet id, region, language, tweet.

As we started the implementation of code. We found out that many tweets were not in English language because of which prediction was not accurate so because of this we filtered out English tweets and did not consider tweets in other language.

2.2 Features:

Bag of Words: In bag of words, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity[i]. A bag of words feature vector consists of all of the words in the article as independent features. In these experiments, all of the words were added to a list and only the top 2000 most frequently occurring words were kept. Each of the words in this list was then compared to the words in the review list, and a dictionary was generated that mapped each of the features to either True or False, denoting whether the feature appeared in the review. This is known as a binary feature vector.

Bigrams: The bag of words feature model, ignores some of the relationships between words that affect their meanings in the context of the article. For example, the phrase '*bad quality*', has a different meaning than *'bad'* and *'quality'* appearing independently. These relationships between words can be captured in the by including bigrams. Using the *'bigram'* function provided by NLTK, all of the bigrams in each of the articles are saved and sorted by frequency. The top 200 of them are included in the feature vectors for each of the articles

2.3 Classification

Naive Bayes: Naive Bayes classifiers are linear classifiers that are known for being simple yet very efficient. The probabilistic model of naive Bayes classifiers is based on Bayes' theorem, and the adjective naïve comes from the assumption that the features in a dataset are mutually independent. In practice, the independence assumption is often violated, but naive Bayes classifiers still tend to perform very well under this unrealistic assumption [1]. Especially for small sample sizes, naive Bayes classifiers can outperform the more powerful alternative. The conditional probability can be decomposed as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}.$$

This can be formally written as

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}}\ p(C_k) \prod_{i=1}^{n} p(x_i|C_k).$$

Support Vector Machines: SVM is a supervised machine learning algorithm which does some extremely complex data transformations, then figures out how to separate your data based on the labels or outputs you've defined. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs. The benefit is that you can capture much more complex relationships between your data points without having to perform difficult transformations on your own. The downside is that the training time is much longer as it's much more computationally intensive.

3. Experiments

We formulate emotion detection as the obvious binary classification problem. The dataset required pre-processing as the positive bias in the training data available from the reviews presented a big challenge. We started our experiments from *AppleIphone 6* but it had a very high polarity of positive reviews that we changed our training product to a more moderate opinion phone *Blu Advance* on Amazon. Even this phone had a total rating of 4.1 stars out of 5 stars which has a high positive bias but we also needed to consider a product with sufficient number of re-

[1] http://www.tweepy.org/

views to be able to train our model efficiently. This product had more than 3600 reviews when we performed crawling on Amazon website.

We used the stars rating on amazon as our labels and converted it to values of zeros and ones to generate a classification problem of binary values. The ratings above 3 were considered positive and labelled with 1 rating, others (1, 2 and 3 rating) were labelled as 0. Later, we found that the positive reviews had a very high frequency so we dropped reviews with rating 4. After all the rearrangement, our dataset had 3096 with 1909 positive reviews. So, after the preprocessing steps, our training dataset had approximately 60% ratio of positive to negative reviews.

We performed initial feasibility experiments using two different classifier types, Naive Bayes and Support Vector Machines. These initial experiments were based only on careful word unigram features from review texts. Performance as measured on the development set ranged from Naive Bayes at 67.0% accuracy to Support Vector Machines at 74.0% accuracy. We performed experiments on different combinations of feature sets and came up with different results. We used 10-fold cross validation to check our accuracy over the entire training set.

After being satisfied with the accuracy of our training classifier, we moved ahead to crawl twitter tweets to make opinion predictions on them. We crawled twitter for tweets related to Iphone6, Note5 and Zenfone. Further processing applied has been explained above (Dataset Section 2.1). Then, we filter those tweets upon the specific feature whose quality we are going to predict. Then, we use our trained classification model on those tweets. The results of different settings we used are summarized in the next section.

4 Results

4.1 Feature sets:

We tried different settings of featuresets on Naïve Bayes Classifier. We first used all the review to collect all the words and used 2000 most frequently used words as our featureset by applying Bag of Words. We used this both on the review 'Text' and 'Title'. This experiment on Amazon text is considered as our baseline algorithm which gave us 64% accuracy.

Then, we analyzed different features. As shown in Figure 1, the accuracy of Title feature was slightly better to 67%. This was as expected because people have a clear opinion in the review title as they like a product or not and it makes it much easier for a classifier to learn.

The accuracy decreased when we used bigrams as our features. We expected improvement but it was disappointing to observe that using Bigrams on the dataset did not performed as expected and the accuracy was decreased considerably. Even combining both of them as feature sets didn't performed better.

Baseline	**64%**
Title Unigrams	67%
Title Bigrams	61%
Text Bigrams	51%
Text + Title Bigrams	65%

Figure 1

4.2 Cross Validation Results

We then used Support Vector Machines classifier and trained our classifier using Bag of Words unigrams only as our feature set. The mean accuracy observed after 10-fold cross validation using SVM was 81.8%.

The accuracy of individual folds were-
[0.6483871, 0.69354839, 0.87741935, 0.90967742, 0.82580645, 0.78064516, 0.93870968, 0.91909385, 0.82847896, 0.76623377]

SVM Evaluation Report				
	Precision	Recall	F1-score	Support
Pos	80%	0.46	0.58	1909
Neg	73%	0.93	0.82	1187
Avg/ total	76%	0.75	0.73	3096

Figure 2

Most Informative Features			
Feature = 'Review Title' Bag of Words	Feature = "Review Text" Bag of Words	Feature ="Review title" Bigrams	Feature = "Review Text" Bigrams
Or	love	(great !)	(start crashing)
Love	waste	(love size)	(pictures clearly)
Bought	reset	(disappointed)	(not good)
Smart	freezing	(excellent mp)	(good choose)
After	error	(after bad)	(first time)

We performed Grid Search to calculate best parameters for SVM using the below parameter grid

```
param_grid=[{'kernel': ['linear'], 'C': [1, 10,
100, 1000], 'gamma': [0.1, 0.01, 0.001, 0.0001]
}, {'kernel': ['rbf'], 'C': [1, 10, 100, 1000], 'gam
ma': [0.001, 0.0001]}],
```

The best parameter selected was (C=100, gamma=0.0001, kernel='rbf').

We created an evaluation report based on our tests as shown in Figure 2

4.3 Most Important Features

Table above shows a list of the most important keywords that were used by our classifier for the classification. We can observe words like *love* is positive and words like *waste, reset, error, not good* indicates negative review.

4.4 Prediction on Twitter Tweets

We applied our trained models on the processed twitter tweets to perform two experiments. First, to estimate the ratio of the polarity of positive to negative reviews as shown in Figure 4. Then, we applied our model to predict public opinions about different characteristics of Zenfone. We describe in Figure 5 the results about the camera, storage, processor, cost and weight of Zenfone. We observe that people seem to be quite disappointed with the processor (speed or processing power) of this phone. The example tweets used to describe them were –

regarding speed(RAM): '*Thats a way to #confuse people with the 4gig RAM model of Zenfone 2'*

regardingcost: '*I dunno why but the specs are good but price is too high.' 'Its a beast in budget. '*

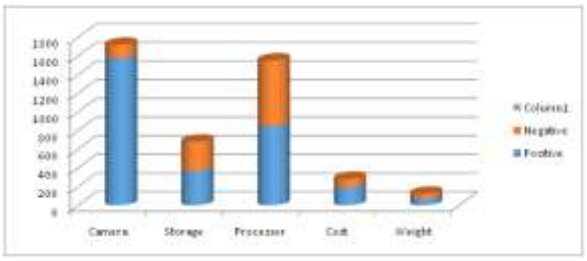

Figure 3: Zenfone Tweets Prediction

5. Conclusion

Generally, the results of this experiment were successful. The classifiers managed to accurately tag a great amount of user-generated data much further past the random baseline of 50%.Naive Bayes performed good on smaller dataset. But, as we increased the size of dataset, SVM outperformed Naïve Bayes in our classification. Our best classifier after parameter tuning performed at the mean accuracy of 81.8 % in 10-fold cross validation. The graphs about the different parts of a cellphone describe public opinions about them. Companies can use this information to analyse what people like in their product and where they are lacking and need more work. We hope this paper fulfil its purpose and can benefit an organization in taking better strategic decisions.

In future work, this work could be extended to make our prediction system more accurate and achieve better accuracy. We can use the concept of SentiWordNet and POS tags to improvise our model. We want to use this model to better predict results on more cell phones and then extend it to a wider category of commodities from electronics to other range of products as well.

6. References

John Blitzer and Hal Daumé, Domain Adapta tion, Paper available at: http://john.blitzer.com/talks/icmltutorial_2010.pdf

Hal Daume III, Frustratingly Easy Domain Ad aptation, Paper available at: http://www.umiacs.umd.edu/~hal/docs/daume07 easyadapt.pdf

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer rviews. In Proceedings of the Tenth ACM SIGKDD International Con ference on Knowl-edge Discovery and Data Mining, KDD '04, pages 168–177, New York, NY, USA. ACM

John D. Burger and John Henderson and George Kim and Guido Zarrella. Discriminating Gen- der on Twitter

ShlomoArgamon, Moshe Koppel, James W. Pennebaker, andJonathan Schler. 2007. Min ing the blogosphere: Age,gender, and the va rieties of self-expression.

John D. Burger and John C. Henderson. 2006. An explorationof observable features related to blogger age. In ComputationalApproaches to Analyzing Weblogs

Analyzing Weblogs: Papers from the2006 AAAI Spring Symposium. AAAI Press.Chris Calli son-Burch and Mark Dredze. 2010. Creating speechand language data with Amazon's Me chanical Turk.

In Proceedingsof the NAACL HLT 2010 Work shop on CreatingSpeech and Language Data with Amazon's Mechanical Turk,CSLDAMT '10. Association for Computational Linguis tics.

Bo Pang, Lillian Lee, and Shivakumar Vaithy anathan. 2002. Thumbs up?: Sentiment classification using machine learning tech- niques. In Proceedings of the ACL-02 Confer- ence on Empirical Methods in Natu-ral La- nguage Processing - Volume 10, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Associa tion for Computational Linguistics.

About Authors :

Sohil Jain is graduate student at Indiana Univer- sity Bloomington. He is completing his Mas- ters's in Data Science. He has three years of work experience with Sears Holding Pvt. Ltd.

Ashutosh Bhargave is a graduate student at Indi- ana University Bloomington. His Master's area of specialization is Data Science. He has two years of work experience with Indian Institute of Technology, Bombay.