Sentiment-based Review System for Product Features

Priyanka Girase, Shweta Phirke, Pragadheeshwaran Thirumurthi
University of California, Los Angeles
{priyanka,sap19, pragadh}@cs.ucla.edu

Abstract

In today's world a number of reviews are available for any product/service and it becomes increasingly difficult and impractical to read every review and choose the required product. Moreover, there is sometimes a need to analyze the product based on the weightage given to its features. When a person wants to use a certain product he may want to know what reviews the product has received and also what other people think about individual features. Hence project aims at presenting an approach for automating the whole process of combining multiple reviews and presenting a concise and useful resource to the user. Minqing Hu and Bing Liu[1] use the method of Apriori algorithm for feature extraction and SentiWordnet as bag of adjectives for determining sentiment orientation of extracted features. Our approach is greatly inspired by their implementation methodology, though we attempt to implement more concepts to improve our results. Considering the human interpretation in sentiment analysis, we choose to perform human evaluation of our experimental results to see how close we are to user expectations or analyze the possible shortcomings in our approach.

Introduction

In the current scenario, if a user has to choose a product based on the reviews available on the product, he has to go through a huge collection of websites & reviews and compare them to find their relevance. The user might read only a few reviews and miss out the important ones. This would result in deciding on a product without analyzing the pros and cons of various features of the product from all the available reviews. Also, presently not many systems concisely analyze multiple reviews of a single feature of a product. This makes it difficult for a user interested in finding about only one particular feature review as lot of human effort and time is wasted in finding an appropriate match for the requirements.

In order to alleviate the problem of browsing through various review sites, a better solution would be to integrate reviews from multiple sites and present a concise picture to the user. This involves taking into consideration the sentiment value of each document. It mainly deals with analyzing the subjective nature of user review for a product and trying to tabularize features against the resulting user opinion. A very good problem that this system resolves is that of evaluating the weight of each and every feature of a product instead of the overall product. For instance, there is a possibility that a single feature of the product is rated extremely bad by many reviews but the overall sentiment of the reviews gives a positive feedback due to the good review of the other features. But a user interested in a product just for some particular feature should have some way to know that even if the product is rated good the feature he is interested in is not so well rated. This problem is solved using the concept of extracting the features and the feature specific sentiments from the entire review.

Related Work

The concept of Sentiment Analysis is most studied topic and many research papers proposing various techniques have been presented. Using Association Rule Mining, Hu and Lui[1] have presented an approach to feature based analysis. In this paper, they enhance the results of Apriori using some Feature Pruning heuristics like compactness pruning and redundancy pruning. Apart from this, their method also includes infrequent feature detection. They perform opinion words extraction based on relative position of word and its associated adjective.

Implementation

Two important sub-tasks involved in our project are:
(1) Identification and extraction of product features
(2) Analysis of the association of extracted opinion words with the extracted features
Stepwise implementation can be stated as below:

1. Data Collection and Pre-processing:

For the purpose of our experiments, we collected huge dataset from Amazon and it covered reviews for many electronic products. The obtained data was in XML format and some pre-processing was done to extract review text into separate text files. Every sentence in the text file represented individual user review.

2. Part of Speech Tagging:

Next step in our implementation was to tag every word in the sentences of review files as corresponding parts of speech. This was needed to identify the possible nouns and adjectives in the sentence. The nouns can intuitively be referred to features of the products while adjective associated with it can be used to give opinion regarding that noun feature. In our implementation we have used Stanford POS tagger to do the work of tagging sentences in review files.

Example: Consider the following review sentence,
This phone comes with superb battery.

The outcome of Stanford POS tagger,
This_DT **phone_NN** comes_VBZ with_IN *superb_JJ* **battery_NN** ._.

Here the bold print represents nouns and italics represent adjective. As can be seen, our intuition holds that nouns mostly correspond to features and adjective represents opinions associated with the features.

3. Feature Extraction:

Next step is to extract the above tagged features. Understanding from Hu and Lui[1], our assumption is that users will most likely criticize or praise features of the product in their reviews. So the frequency of occurrence for such words will be high. Hence, we used Apriori Algorithm to mine such frequently occurring words (nouns). We run the algorithm as a Python script. The input to this is our file containing only those words from the reviews which are tagged as NN/NNS/NNP. The features obtained from this are used as bag of features for sentiment analysis to be performed on.

Challenges in Feature Extraction:

Basic run of Apriori algorithm will unintelligently mine frequently occurring words (nouns) without considering a two-word feature as a single feature. To elucidate, consider the same statement above with small modification,

This phone comes with superb battery life.

This_DT **phone_NN** comes_VBZ with_IN *superb_JJ* **battery_NN life_NN** ._.

Simply extracting nouns as features will give 'battery' and 'life' as separate features and eventually associate same adjective to both of them when in reality 'battery life' should be considered as a single feature. For overcoming this problem, we used bigrams which associate consecutive occurring nouns as a feature. The intuition is that consecutively occurring frequent noun-noun pairs are features. In the later section, we show that our intuition holds and we do get such pairs as features.

Now consider a second scenario with following set of review sentences:

The pictures are absolutely amazing, the pictures are fantastic, the pictures are just plain superb.
….
The zoom is ok.
…
It's zoom is not bad.
…
The zoom is good.

Basic run of apriori will associate a frequency of 3 with 'pictures' as well as 'zoom', whereas only the latter is the correct case. This consideration is important because, as we evaluate in later section, using apriori we are considering top k frequently occurring words as features. Other words which are not so frequent are observed to be irrelevant words and not product features.

4. Sentiment Analysis:

To analyze the semantic orientation of the extracted features we perform sentiment analysis using SentiWordnet. SentiWordnet acts as a bag of adjectives for us and given an adjective it returns positive and negative scores associated with that adjective. The features extracted from above process are passed to Python script which runs SentiWordNet using NLTK.

Example (3): The size is comfortable and convenient.

The scores associated with every adjective: Comfortable → pos: 0.625 neg: 0
Convenient → pos: 0.625 neg: 0.25
We calculate the sentiment value of the feature 'size' using following formula:
N = Nouns, F = Feature, S = Sentiments
For all N as F in i,

$$\mathbf{S(F) = \sum Adj_{ij}^{pos} - \sum Adj_{ij}^{neg}}$$

where i → review statement no.
j → adjective number in that statement

Hence considering this formula, the sentiment score associated with 'size' is measured to be 1. Likewise, we have considered every occurrence of "relevant" feature in every statement and summed up all the scores associated with it.

In addition to using plain SentiWordNet, we have taken 'negation' in the sentences under consideration. Words like 'not', 'isn't', 'no', 'hasn't' coming with any adjectives negate the meaning of that adjective. Hence it is necessary to improve this otherwise completely opposite sentiment would get associated with a feature. We implemented this concept by running programs in Python script.

Experiments on Amazon Dataset

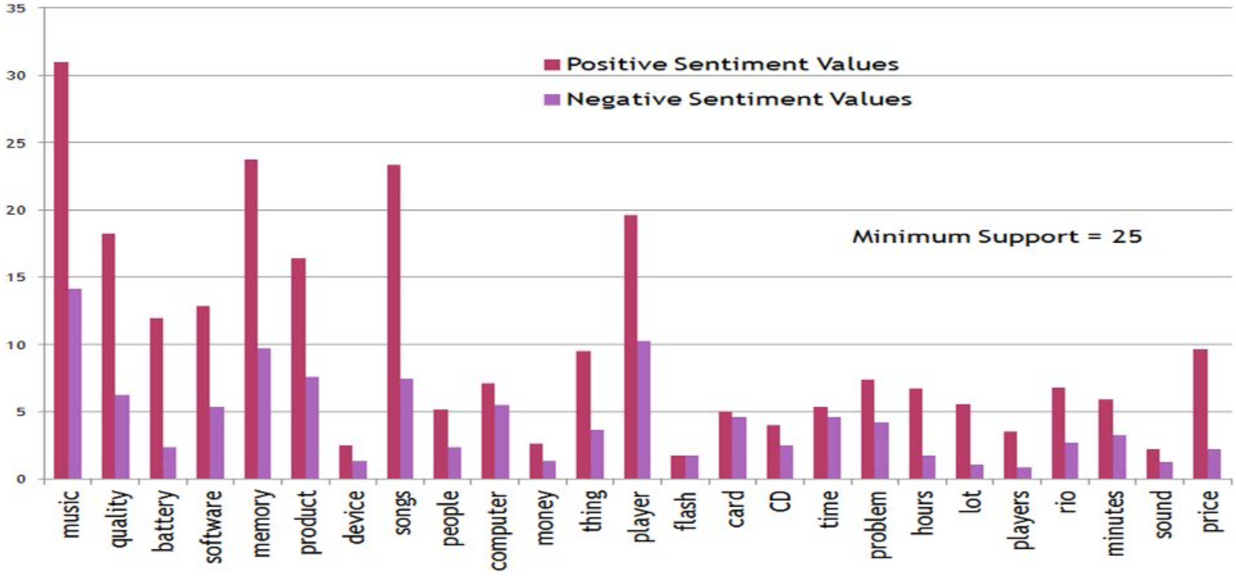
For performing some modifications to basic Apriori algorithm, we used a very huge dataset of over 500 reviews for an electronic product: Rio PMP 300 MP3 Player.

We implemented two runs of Apriori Algorithms:

(1) Applied fixed minimum support on the whole dataset
(2) Partitioned the dataset into small parts and applied same minimum support for all parts

The results for both are shown below. The first run corresponds to basic Apriori. We observed that the results obtained as features consisted of many irrelevant words (which were not actually product features). As can be seen after first run, we get results like 'hours', 'time', 'minutes' which are actually not required and relevant features.

The reason for considering next run is because choosing a single minimum support for the whole dataset did not give relevant set of features. Using lower value of minimum support returned many irrelevant features whereas using higher value for minimum support removed relevant and required features. Both these cases are not desired as our aim is to stabilize precision as well as recall.



Figure(1): Fixed minimum support applied to whole dataset

It is expected of smaller sized datasets to return high precision value than recall. Based on this, our intuition is that partitioning the size of data into smaller parts and performing the Apriori runs on each of such parts will improve the precision. This is because the frequency of a feature occurring in a subset increases when compared against the total features in that subset. Hence using fixed minimum support for partitioned dataset, we got the following result. As can be seen, there is significant improvement in the precision though some irrelevant features are still returned.

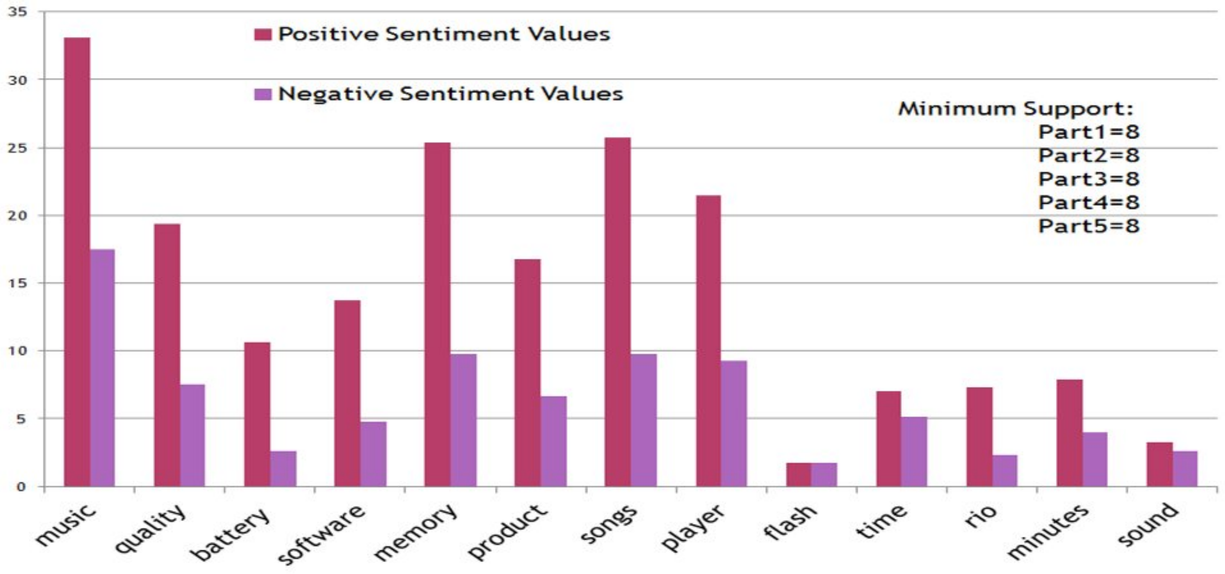


Figure (2): Fixed minimum support applied to partitioned dataset

Rio MP3 player	No. of annotated features	No. of extracted features	No. of relevant features from extracted	Precision	Recall
Case 1	28	25	15	0.60	0.53
Case 2	28	13	10	0.77	0.36

Table (1): Precision and Recall for 2 Apriori cases

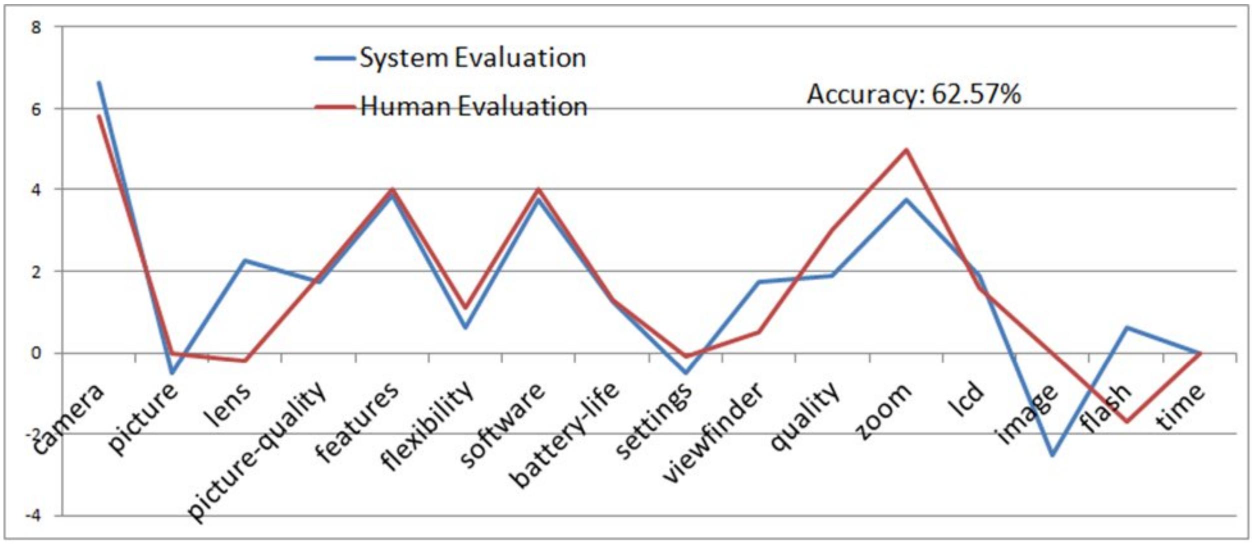
As can be seen from Figure (1), the number of features returned by the system is very high as we are applying same minimum support to entire dataset. The precision in the case 1 is around 60% whereas the recall is less and around 53%. The reason is we have not considered infrequent feature mining which would have improved the overall recall.

By partitioning the dataset and then implementing the apriori run on each part, we have shown to improve the precision to around 77%. As can be seen from Figure (2), the system still returns irrelevant words as features like ‘rio’ and ‘minutes’. The other reason is also that the dataset considered mentioned these terms a lot many no. of times and Apriori was not that successful in mining relevant features as it only mines frequent terms.

Evaluation

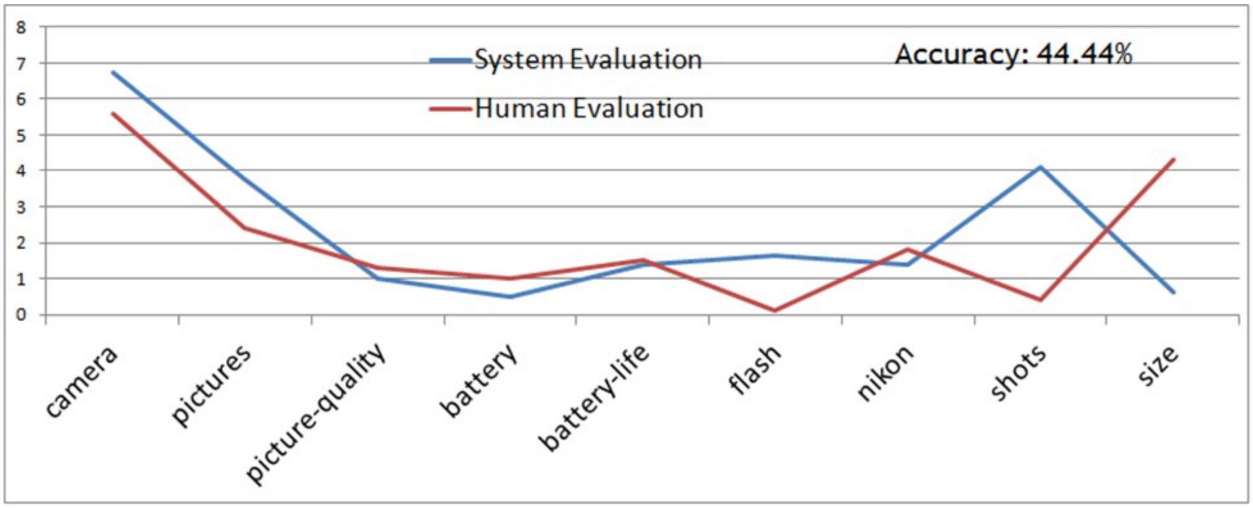
The results for above mentioned modifications to Apriori algorithm were observed along the graph itself. However, considering it is a huge dataset, performing user evaluation was very difficult. But we evaluated the results of sentiment analysis against users' opinions and derived the following graphs. To draw appropriate conclusions from our results, we evaluated for 3 different datasets: Nikon camera, Canon camera and Nokia phone.

The graphs below show that our results are somewhat closer to user evaluation for most of the features.



Figure(4): Evaluation graph for Canon camera

On calculation, the accuracy for graph in Figure (4) comes around 62.57%. As can be seen, the sentiments are opposite in sense for some features like 'lens'. On observation it is found that this is because the statements talked negative about some other feature but the adjective got associated with the 'lens' feature as well. Also, some irrelevant feature like 'time' is included as a result of Apriori which mines only frequent items.



Figure(5): Evaluation graph for Nikon camera

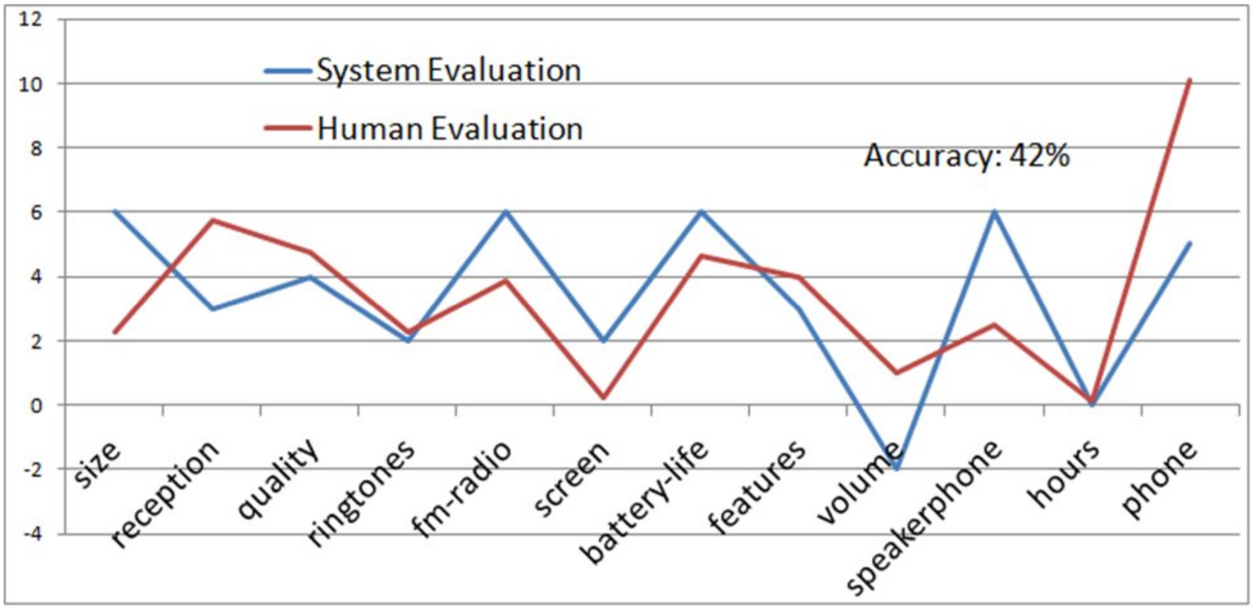


Figure (6): Evaluation graph for nokia cell phone

Since sentiment analysis requires a human interpreter with some domain knowledge, we evaluated our system by manually reading all the reviews. For graphs in Figures (5) and (6), the accuracy is around 44.44% and 42% respectively.

In Figure (5), the accuracy is less because considering all the features mined by Apriori returned many irrelevant results also. The sentiments associated with these were next to nothing but the overall accuracy was much lesser than this. Hence, this again leads us to conclude that it is necessary to mine infrequent features by considering point-wise mutual information.

In Figure (6), the results are skewed for some features like 'size' and 'reception'. For review statements like 'size is small', the adjective 'small' holds a lot of negative score as per the SentiWordNet dictionary whereas when it comes to electronics, the smaller the item is the better it is. This was the sentiment expressed in all the statements. Also, for 'reception' certain adjectives associated were 'top-notch' and 'cool' whereas these words are not included in that lexical resource.

Another observation, our system is not intelligent. For statement like '…blocks the lens', our system cannot sense the negative sentiment because there is no such adjective associated with 'lens'. This is another reason for getting skewed results and decreased accuracy.

Also, our dictionary does not include scores for slang words like 'cool' and 'hats off' and others. But we realized that this is the jargon mostly used by people while commenting any review.

Thus overall observation, suggest human evaluation is of utmost importance when it comes to sentiment analysis.

Conclusion

From Apriori implementation approach we can conclude that partitioning the dataset and then applying minimum support somewhat improves the precision to around 70%. We got an average on 50% accuracy from the sentiment analysis provided by our system measured for 3 different datasets.

Some of the results obtained from sentiment evaluation were opposite to that of human evaluation. Reasons for this are:

1. SentiWordNet associates negative feelings with words like ‘small’, ‘free’, ‘loud’, ‘cool’ whereas the review statements expressed positive feelings. We conclude that some human interpretation is always needed in this domain to analyze this.
2. Our system did not consider point wise mutual information between noun and the associated adjective and hence for some review statements which had noun-positive adjective and noun-negative adjective pairs, the result got skewed because of wrong associations.

We encountered the problem of slang words and ambiguous statements in our dataset and associating appropriate sentiment with features became difficult.

Future Work

One of the many improvements to sentiment analysis would be to consider adjective-adverb combinations instead of just adjectives. Since sentiments of users are amplified by adverb usage an optimized enhancement would be to include their scores as well.

Secondly, the process of feature extraction can be enhanced effectively to give high precision and recall using advanced classifiers like Support Vector Machine (SVM)

Some techniques for considering mutual distance between sentiment and feature such as Pointwise Mutual Information (PMI) may improve the accuracy of the results. This can also be used to mine infrequently occurring features.

Another extension would be to compare the results obtained in our method with manufacturer’s claims. This would be an interesting find to know whether manufacturer is living upto user’s expectations.

References

[1] Minqing Hu and Bing Liu, "Mining Opinion Features in Customer reviews"

[2] Bo Pang and Lillian Lee, "Opinion and Sentiment Analysis"

[3] Kerstin Denecke, "Using SentiWordNet for Multilingual Sentiment Analysis"

[4] Ana-Maria Popescu and Oren Etzioni, "Extracting Product Features and Opinions from Reviews"