Opinion Mining on Customer Product Reviews Using Supervised Machine Learning Techniques

Sivakumar A*, Jagadeesh Babu S*, Sathya Vignesh R*, Shyam M*, Yogapriya J**

** Assistant Professor, ECE, R.M.K. Engineering College, **Program Analysist, CTS, Chennai*

Abstract: In last decades online product sale is increased. The customers want to buy a quality product is very difficult in recent year. After buying only we know the problems in the product. After lancing many months users buying the product with problems. But many users put their Opinion in the review pages. Customers are very difficult to find the best product. Opinion Mining (OM) is the best tool for selecting the best product. OM on Product reviews refers to the process of analyzing the sentiment associated with it. This paper discussed about an attribute – level sentiment analysis of the product was done and also performs a three – class classification

Keywords: Opinion Mining, Feature Extraction

1. INTRODUCTION

Opinion mining (also known as reaction investigation or feeling AI) refers to the apply normal language dealing out, wording examination, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information [1][2]. Reaction Investigation is generally useful to say the purchaser materials such as reviews and study of outputs, Internet and community portal, and healthcare products for purposes that range from advertising to buyer check to clinical drug [3]. Rather than performing an overall sentiment analysis, this paper focuses on an aspect – level Opinion Mining which is in other words also called as Feature – level Sentiment Analysis or Attribute – Level Sentiment Analysis. This is done in three major phases, also called as three major modules – product attribute extraction, attribute phrase extraction and performing sentiment analysis or opinion mining on those phrases that are associated with those extracted attributes. This is of major concern as a user might not be interested in a feature set which other users might be interested in. It involves the use of Supervised Machine Learning Techniques where the system is trained to fit the training dataset to come with a model that can then exist used to predict the values of the target variables [4]. The Machine Learning Algorithms used here includes Logistic Regression, Support Vector Machines (SVMs) and Random Forest Classifier [5]. The proposed system tries to overcome the drawbacks of the existing systems by providing a graph of overall sentiments and listing the features of a product and their feature – level analysis of sentiments on customer product reviews. First step of any Machine Learning process is Data processing.

2. DATA PREPOCESSING

Data extraction is means, the group and treatment of items of data to create significant information. In this intelligence it can be measured a division of information extraction, the alter information in any method measurable by a viewer [6]. The phrase Data Extraction has also been utilized in the past to refer to a section within a group answerable for the function of data extraction purposes.

- Data extraction possibly will involve a variety of procedures, together with:
- Validation – Ensuring that given input is correct and applicable
- Sorting – "Alignment of data in some order and/or in separate groups."
- Summarization – Reduction of complete information to its important keywords.
- Aggregation – Grouping different parts of information.
- Analysis – "Gathering, grouping, examination, understanding and arrangement of Information."

- Reporting – catalog feature or abstract information
- Classification – Break ups information into different classes.

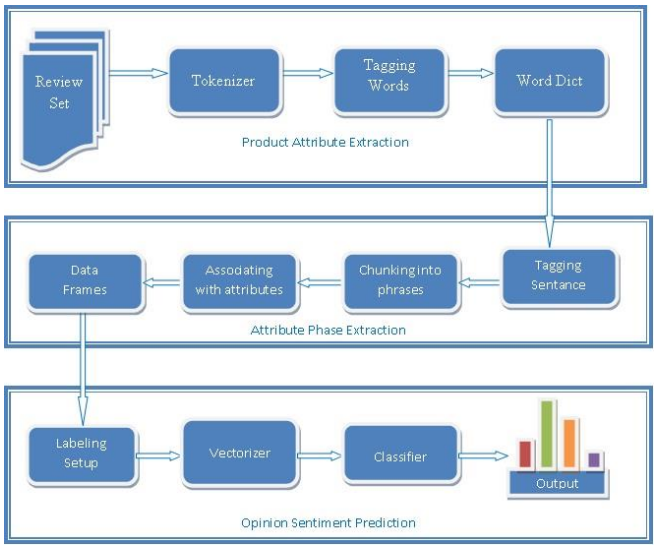

Fig. 1. Opinion Sentiment Prediction Block diagram

3. PRODUCT ATTRIBUTE EXTRACTION

One of the important process is called entity detection is product attribute extraction. Automatically preparing product attributes from words assists E-commerce business appropriately progression customer inquiry and equivalent it the correct products [7][8]. Advertisers desire to identify what public is chatting concerning correlated to their products. Company Analysts' needs to search out what products are fashionable based on consumer thought.

Product attributes are specific properties related to a product. For example, Apple iPhone 5 with black color has 4 major attributes: Company name as "Apple", brand name as "iPhone", generation as "5", and color as "black". A complete set of attributes define a product.

What challenge do we face in extracting product attributes? We can create a dictionary of products and their corresponding attributes, but simply relying on a dictionary has its limitations:

A dictionary is not complete as new products are created. There is ambiguity in matching the attributes. Misspelling, abbreviations, and acronyms are used often, particularly in social media such as Twitter [9].

The reviews are tokenized into different sentences and tagging individual words in it. A manual inspection of different review sentences showed that most of the product attributes were either nouns, adjectives, adverbs or a combination of them. We then extracted only those word phrases which used the patterns mentioned in the following pseudo code and occurred a certain number of times. Regex grammar is used for tokenizing to extract noun phrases [10].

P1: {<JJ. *|NN. *><NN. *>+} # battery charging system

P2: {<RB|RBR|RBS><JJ. *|NN. *>+} # Shows noun doing action

P3: {<NN. *>+} # battery life, lcd screen

Many of these extracted phrases were synonyms of each other. In order to club them into similar groups, we used the following three properties:

- Common Words: Attribute expressions sharing some common words are likely to belong to the same group. For example, "battery life", "battery", "battery charger" etc.
- Lexical Similarity: Attribute expressions whose words are synonymous in WordNet [11, 16, 23, 33, 36] are likely to belong in the same to the same group. For example, "battery" and "charger"
- Domain Filtering: Attribute expressions whose words are related with the application domain or product are likely to be most relevant product attributes. For example, "screen", "internet", "camera" are essential features of a cellphone and hence relevant.

Keeping these constraints in mind, we proceeded with the following:

i. We formed a dictionary (called "word_dict") with words as keys and a list of all extracted attribute phrases containing that word as values.
ii. From these dictionaries, we related every word key with every other word key if their corresponding phrase list had noun-based lexical similarity in WordNet above a particular threshold (0.8 in this case).

We now had a list of tuples of words which were similar to each other due to the phrases in which they were contained. From this list, we wanted to group together all the word tuples which had a common word. Thus, we formed another dictionary (called "topics") with key as the common word and value as set of all the words in the tuples containing that common word.

The dictionary "topics" contained many words unrelated to phone attributes such as "hour", "work", "good", "simple", "expectation", "fine" etc. Thus, we filtered out those words keys which did not have a noun-based lexical similarity with phones in WordNet [11].

4. ATTRIBUTE PHARSE EXTRACTION

Initially, we tried to associate the entire sentences to the product attributes they contained, under the assumption that usually customer reviews are written by regular people and are hence, simple in structure - reviews describe one product attribute per sentence [12][13]. But on analyzing the training / development data set, we found that this was not the case. Each sentence in the review were mostly complex, with multiple product attributes and multiple set of descriptors.

We classified the complex sentences into the following five categories:

Sentences with a single product attribute, and a single set of descriptors.
E.g. ‘The battery life is very bad.”

Sentences with multiple product attributes, and a single set of descriptors.
E.g. “The camera, as well as the processor is very good.”

Sentences with a single product attribute, and multiple sets of descriptors.
E.g. The screen is not so good, but it is enough for general usage.

Sentences with multiple product attributes, and multiple set of descriptors.
E.g. “I hate the camera, the screen is also bad, however the battery is good.”

Sentences with pronouns, and associated set of descriptors.

E.g. “It has a 13 Megapixel camera. It takes really good pictures.”

In order to overcome these challenges, and correctly extract single attribute phrases from the above 5 type of sentences, we tried using different algorithms to check which one provided the best results.

The implementation can be divided into three stages:

Tagging the sentences
Chunking / Tokenizing the sentences into phrases
Associating phrases with the respective product attribute

A. *Tagging the Sentences*

We implemented a trigram backoff tagger, which was trained on brown corpus categories of ‘news’, ‘editorial’, and ‘reviews’ since they were similar to our product reviews [14][15].

In order to align the tags of some phrases / terms (like "not") to our liking, we added some additional training data sets.By doing so, we were able to appropriately tag the set of adverbs and conjunctions present in the text, which was not being done by either the “nltk.pos_tag” or by the trigram back-off tagger with brown corpus training data set [16].

B. *Tokenizing the sentences into phrases*

To correctly chunk the sentences into single attribute phrases, we use grammar rules to identify the associated descriptors for each product attribute in the sentences, and tokenize them appropriately.

Following are the algorithms which we used:

Context Free Grammar Algorithm: This algorithm utilizes "grammar", which specifies which trees can represent the structure of a given text. These can then be used to find possible syntactic structures for the sentences.

- Using Regex Parsers to tokenize sentences: In this we tried to find patterns in the tags of the phrases, and then define regex patterns to try and chunk them.
- Utilizing conjunctions, and other punctuations in the sentences to perform tokenization: In this, we tried using the conjunctions and the punctuations present in the sentences to determine distinct phrases.

C. *Associating phrases with product attributes*

The extracted “most talked about" product attributes and split the review sentences in phrases containing a single product attribute, we are left we the job of associating each phrase with the corresponding product attribute it contains. We need to build this association so that the sentiments predicted per phrase can be summed-up for the corresponding product attribute [17]. Phrases not containing the extracted product attributes will be discarded.

To do this, we first create a data frame for every extracted product attribute. We then use a simple algorithm to traverse through each phrase and find whether it contains a product attribute or its synonyms. If yes, then the phrase is added to corresponding attribute's data frame. If a phrase contains multiple product attributes then the phrase is added to the data frames of all the contained attributes

.

5. SENTIMENT PREDICTION

Sentiment analysis (sometimes known as opinion mining or emotion AI) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information [18]. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine [19].

Generally speaking, sentiment analysis aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event. The attitude may be a judgment or evaluation (see appraisal theory), affective state (that is to say, the emotional state of the author or speaker), or the intended emotional communication (that is to say, the emotional effect intended by the author or interlocutor).

In order to extract sentiments from the phrases derived in the Product Phrase Extraction and Association step, we decided to use various machine learning techniques. We used vectorizers to extract and learn syntactic patterns accompanying different sentiments in the reviews. We then trained classifiers on manually labelled training dataset to predict the sentiment contained in each of the phrases.

All submitted paper should be cutting edge, result oriented, original paper and under the scope of the journal that should belong to the engineering and technology area. In the paper title, there should not be word ‘Overview/brief/ Introduction, Review, Case study/ Study, Survey, Approach, Comparative, Analysis, Comparative Investigation, Investigation’.

In order to extract sentiments from the phrases derived in the Product Phrase Extraction and Association step, we decided to use various machine learning techniques. We used vectorizers to extract and learn syntactic patterns accompanying different sentiments in the reviews. We then trained classifiers on manually labelled training dataset to predict the sentiment contained in each of the phrases.

Thus, this step can be separated into three segments:

1. Manually labelling the training set into positive, negative and neutral sentiments
2. Identifying the best vectorizer to extract characteristic features differentiating one sentiment from another
3. Identifying the best classifier which will used these characteristic features to predict sentiments

This involved finding which combination of vectorizer and classifiers works the best for the dataset, without over-fitting to the training dataset.

D. *Labelling Training Set*

Using the procedures explained in the Data pre-processing section, we split the Amazon review sentences into single attribute phrases. Then we manually labelled the data, classifying them into ‘Negative mentions’ (represented as -1), Neutral Mentions (represented as 0), and Positive mentions (represented as 1).

E. *Vectorizer*

We tried various vectorizers found in SciKit – Learn python machine-learning library with different parameters values before finalizing which vectorizer will be the best fit for our dataset.

Following are the vectorizers we considered, along with the variations in parameters we did.

Count vectorizer converts text 'documents' into a matrix of token counts, specifically, a sparse matrix representation of the counts using the scipy.sparse.coo_matrix. We specified the analyzer to 'words', allowed lowercasing of tokens, and also asked the vectorizer to consider Unigrams, Bigrams, and Trigrams as features. We limited the number of features to 1000.

Tf-Idf Vectorizer converts the collection of raw 'documents' into a matrix of TF-IDF features. It is equivalent to applying Count Vectorizer on the set of 'documents' and then applying Tf-Idf transformation on it. "Tf–Idf, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus ". In this vectorizer as well, we tried different combinations of various parameters, to output the best results.

F. *Classifier*

To perform sentiment analysis on the phrases associated with different product attributes, in conjunction with the vectorizers, we tried three different classifier algorithms to analyze which combination provided the best precision, and re-call, without over-fitting to the training data set.

Logistic Regression Classifier model is a discriminative model, which can be used to directly estimate the probability of occurrence of y given occurrence of x (p(y|x)). For this model to work correctly there is no restriction on the features to be co-related. It can be used to provide multi-category classification in cases where the categories are exhaustive, and mutually exclusive, i.e. every instance belongs to one, and only one category.

SVM Classifier is also a discriminative classifier, which is formally defined by a hyper-plane, i.e. provided labelled training data set, this classifier outputs an optimal hyper-plane, which can then we utilized to classify new examples.

Random Forest Classifier is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default).

6. PERFORMANCE EVALUATION

During the first conceptual phase of a program, critical business processes are identified. Typically they are classified as critical based upon revenue value, cost savings, or others assigned business value. This classification is done by the business unit, not the IT Organization.

High level risks that may impact the system performance are identified and described at this time. An example might be known performance risks for a particular vendor system.

Finally, performance activities, roles, and deliverables are identified for the Elaboration phase. Activities and resource loading are incorporated into the Elaboration phase paper plans.

G. *Attribute Extraction*

In order to test our algorithm's precision, we manually inspected all the reviews in test data and drew out the most common product attributes being talked about. We considered these as baseline attributes as they were common attributes across reviews of other cellphones as well. We then drew out a list of attributes given by our proposed algorithm and compared them as shown in the Table 4.1.

Table 4.1: Expected Features v/s Actual Features obtained for various products

Manual Inspection	Iphone 5s (test data)	Galaxy S5	Nexus 5
battery	battery	battery	battery
screen	display	monitor, display	display
charger	charger	charger	
speakers	headphone		
camera	camera	camera	camera
keyboard	button, home	button, home	home
software	system	system	
size	light ,lighter		light ,lighter
connection	connection	wireless	
hardware	hardware	resistance, sensor, port, scanner	processor, hardware
body	back	back	
price			
experience			
speed			
	fan		fan

Our algorithm does a pretty good job of identifying the various product attributes with no manual intervention. The precision is about 78.57% on test data set and 64.28% on Galaxy S5 data set. It has surprisingly good precision on Nexus data set which has very few reviews (226 in total).

This is comparable to the baseline precision range of 69-75% given by "Hu and Liu's" work2. The comparatively low precision in Galaxy S5 and Nexus 5 can be due to the fact that our algorithm was unable to capture indirect/derived product attributes such as "price", "speed", "user experience" etc. This is because in our algorithm, the product attributes are purely derived from the words present in the reviews and any other word is not used. So in the case of "speed", reviews will contain words like "fast", "faster" or "slow" and in the case of "price", reviews will contain words like "cheap", "expensive", "rip off" etc. Because the algorithm is not able to club them under one group, the frequency of such words (like "cheap", "expensive", "rip off") is not high enough to be captured.

H. *Attribute Phrase Extraction*

The method to split sentences on conjunctions and punctuations, though simple, gave surprisingly good results on the product reviews. Most of the phrases contain a single attribute with a single set of descriptors. In other words, the resulting phrases are complete in themselves and sufficient to extract sentiments for each of the product attribute they contain as shown in Table 4.2.

7. Opinion Sentiment Predictions

On the training dataset, the combination of Count Vectorizer and Logistic Regression Classifier gave a precision of 81%. However, running this combination on the test dataset made us realize that Count Vectorizer and Logistic Regression Classifier were over-fitting on the training data set and not producing as good results as expected.

On the other hand, though the combination of Tf-Idf Vectorizer and Random Forest Classifier was giving only 76% precision on the training data set, it was giving much better results on the test dataset.

Hence, we decided to trade-off on lower accuracy on training set for better results on the test data set and finalized on using Tf-Idf Vectorizer in combination with Random Forest Classifier to predict sentiments. The precision and recall for Tf-Idf Vectorizer and Random Forest Classifier combination is shown in Table 4.3

Table 4.2: Original reviews and the phrases extracted for product attributes

Reviews	Single attribute Phrases	Associated Product Attribute
I was very excited when I received the two phones. Fast shipping and everything, however after about two months of using the phones white lines appeared on the screen of both phones which then cost me additional money to replace the screens. Very disappointed since I selected new phones and not used ones for the purpose of avoiding additional cost.	I was very excited when I received the two phones	Phone
	Fast shipping	
	Everything	
	After about two months of using the phones white lines appeared on the screen of both phones which then cost me additional money to replace the screens	Screen
	Very disappointed since I selected new phones	Phone
	Not used ones for the purpose of avoiding additional cost	

Table 4.3: Precision and recall for the final combination of Tf-Idf Vectorizer and Random Forest Classifier

```
# Classifier report for the Random Forest Model
print(classification_report(y_dev, y_rf_pred))
```

	precision	recall	f1-score	support
-1	0.76	0.76	0.76	17
0	0.65	0.75	0.70	20
1	0.85	0.74	0.79	23
avg / total	0.76	0.75	0.75	60

8. RESULT AND DISCUSSION

We ran our algorithm on three datasets: Training/Dev dataset of I-Phone 5s (1141 reviews), large test dataset of Samsung Galaxy 5s (2880 reviews) and a smaller dataset of LG Nexus 5 (226 reviews).

Our Training / Dev dataset was manually tagged for sentiment analysis. Hence, we were able to measure the precision and recall for this dataset. On an average, the precision obtained was 0.76, with precision being as high as 0.85 for positive sentiments and as low as 0.65 for neutral sentiments. The recall obtained was on an average 0.75, with little variance across all the sentiments. The recall obtained was very consistent. The overall f-1 score was 0.75.

In terms of product attribute extraction, while tagging the training data manually, we made a list of attributes we were expecting from the algorithm. The extracted attributes for all three data sets are as shown below in the Table 4.4.

Table 4.4: The extracted attributes of the data sets

Manual Inspection	Iphone 5s (test data)	Galaxy S5	Nexus 5
battery	battery	battery	battery
screen	display	monitor, display	display
charger	charger	charger	
speakers	headphone		
camera	camera	camera	camera
keyboard	button, home	button, home	home
software	system	system	
size	light ,lighter		light ,lighter
connection	connection	wireless	
hardware	hardware	resistance, sensor, port, scanner	processor, hardware
body	back	back	
price			
experience			
speed			
	fan		fan

Some of the expected attributes were not obtained in the test data. But on closer inspection we found that this was because these attributes were not frequently talked about in the reviews for the test data.

For I-Phone 5s (training data) the overall results were as shown below in the Figure 4.1. As seen in the graphs, neutral mentions are included while talking about individual product attributes but not when talking about the product overall. This is because, most of the neutral mentions about the overall product were not sentiments, but general statements that do not convey sentiments like "I gifted this phone to my wife", "I travel a lot with this phone" etc.

Hence, we removed the neutral sentiments from the final results for the phone as a whole. Overall, 441 users were happy with the device, while 335 were not as happy, conveying that overall users are not as happy as one would expect. Battery and Charger were the biggest pain points for I-Phone 5s. The camera, the weight and the system were a plus for it.

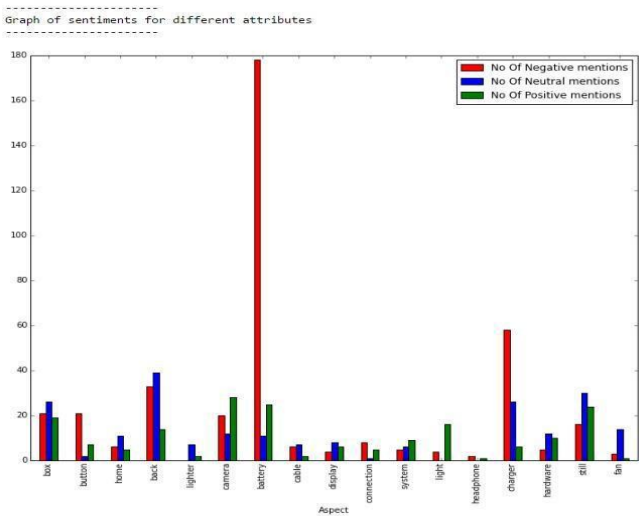

Fig: 2

The results were pretty good in both the test data sets. Samsung Galaxy S5 was evaluated with 2880 reviews and the graph is shown in Figure 4.2. Overall, 1937 positive mentions were obtained with 1005 negative mentions, pointing to overall a better sentiment from the users towards the phone. Charger and Battery were again big pain points along with one of its ports getting negative mentions. The camera and the display are being the positive attributes of the phone.

The results on a test dataset of LG Nexus 5 with relatively less reviews (226 reviews) were also very good. Overall, there were 189 positive mentions along with 99 negative mentions, indicating an overall positive sentiment. Battery and Charger again are getting negative reviews (Smartphones).

However, LG Nexus got positive reviews in many categories like camera, weight, display, processor and hardware as shown in Figure 4.3

9. Conclusion

The algorithm worked well within the product category of Cell phones. The overall precision and accuracy obtained was good, both in terms of feature extraction as well as sentiment analysis.

Every part of the algorithm was automated, and hence the results were obtained with minimum human intervention. The algorithm worked well even when the numbers of reviews were less. The entire algorithm was written in existing libraries and packages within NLTK and hence is affected by the performance of these libraries and packages.

10. FUTURE ENHANCEMENT

The overall results for the paper were very good, but there is always room for improvement. We were able to extract very good features by selecting features that were lexically more similar to the product category ("cellphones") with the help of WordNet. WordNet however, sometimes does not give an accurate measure of how close words are to each other. Substituting it with another resource that can establish a more accurate lexical similarity can maybe provide better results in terms of feature extraction. Creating ontology for products and its attributes is another way of achieving a higher accuracy.

The words are grouped together based on their similarity to each other and clubbed together as one product attribute, and then the attributes are filtered based on their similarity to the actual product category. This process is performance intensive, and hence there is a scope for improvement in terms of performance. Substituting WordNet with some other resource, creating a manual ontology, manually intervening to club potential product attributes into final product attributes etc. are some of the ways in which this can be achieved.

REFERENCES

1. Abinaya R, Aishwaryaa P, Baavana S, Thamarai Selvi N D, Automatic Sentiment Analysis of User Reviews, 2016.
2. Erik Cambria, Soujanya Poria, Rajiv Bajpai, Bjorn Schuller, SenticNet 4: A Semantic Resource for Sentiment Analysis Based on Conceptual Primitives, 2016.
3. Apoorv Agarwal, Vivek Sharma, Geetha Sikka, Renu Dhir, Opinion Mining of News Headlines using SentiWordNet, 2016.
4. Xing Fang, Justin Zhan, Sentiment Analysis using Product review data, 2015.
5. A Jeyapriya and C S Kanimozhi Selvi, Extracting Aspects and Mining Opinions in Product Reviews using Supervised Learning Algorithm, 2015.
6. Alexandra Cornian, Valentin Sgarcian, Bogdan Martin, Sentiment analysis from product reviews using SentiWordNet as lexical resource, 2015.
7. Kang Liu, Liheng Xu, and Jun Zhao, Co-Extracting Opinion Targets and Opinion Words from Online Reviews Based on the Word Alignment Model, 2015.
8. Ankita Srivastava, Dr. M P Singh, Prabhat Kumar, Supervised Sentiment Analysis of Product Reviews using K-NN Classifier, 2014.

9. Malhar Anjaria, Ram Mohan Reddy Guddeti, Influence factor based opinion mining of Twitter data using supervised learning, 2014.
10. S J Veeraselvi, C Saranya, Semantic orientation approach for sentiment classification, 2014.
11. Zheng-Jun Zha, Jianxing Yu, Meng Wang and Tat-Seng Chua, Product Aspect Ranking and Its Applications, 2013.
12. Basant Agarwal, Vijay Kumar Sharma, Namita Mittal, Sentiment classification of review documents using phrase patterns, 2013.
13. Bing Liu, Sentiment Analysis and Opinion mining, 2012.
14. Sadhasivam, Kanimozhi SC, Tamilarasi Angamuthu, Mining Rare Itemset with automated support threshold, 2011.
15. David Garcia and Frank Schweitzer, Emotions in product reviews – Empirics and models, 2011.
16. Alexander Hogenbom, Paul Iterson, Bas Herschoop, Flavious Frasincas, Uzay Kaymak, Determining negation scope and strength in sentiment analysis, 2011.
17. Jantima Polpinij and Aditya K. Ghose, An Ontology-based Sentiment Classification Methodology for Online Consumer Reviews, 2008.
18. S. B. Kotsiantis, Supervised Machine Learning: A Review of Classification Techniques, 2007.
19. Kanayama, Hiroshi, Tetsuya Nasukawa, Fully automatic lexicon expansion for domain-oriented sentiment analysis, 2006.