Rule Based System for Enhancing Recall for Feature Mining from Short Sentences in Customer Review Documents

Tanvir Ahmad
Department of Computer Engineering,
Jamia Millia Islamia,
New Delhi, India
tahmad2@jmi.ac.in

Mohammad Najmud Doja
Department of Computer Engineering,
Jamia Millia Islamia,
New Delhi, India
ndoja@yahoo.com

Abstract - This paper discovers rules for enhancing the recall values of sentences containing opinions from customer review documents. It does so by mining the features and opinion from different blogs, news site, and review sites. With the advent of numerous web sites which are posting online reviews and opinion there has been exponential growth of user generated contents. Since almost all the contents are stored in unstructured or semi-structured format, mining of features and opinions from it has become a challenging task. The paper extracts features and thereby opinions sentences using semantic and linguistic analysis of text documents. The polarity of the extracted opinions is established using numeric score values obtained through Senti-WordNet. The system shows that normal rules discovered earlier are not sufficient to improve recall values as some of the opinions does not contain sentences which are linguistically correct but they express the main idea what the writer wants to convey about his opinion on a particular product. Our experiment uses a method which first identifies short sentences and then uses rules which can be applied on those sentences so that the recall values are enhanced. The paper also applies rules on sentences which are linguistically and syntactically incorrect. The efficacy of the system is established through experimentation over customer reviews on four different models of digital camera, and iPhone.

Keywords-Opinion mining; Opinion analysis; Sentiment Analysis; Text mining; Rules generation; Natural language processing.

I. INTRODUCTION

The World Wide Web is growing exponentially both in terms of size and diversity in the types of services and contents provided. In recent past, due to existence of numerous forums, discussion groups, and blogs, individual users are participating more actively and are generating vast amount of new data – termed as *user-generated contents*. These new Web contents include customer reviews and blogs that express opinions on products and services – which are collectively referred to as customer feedback data on the Web. As customer feedback on the Web influences other customer's decisions, these feedbacks have become an important source of information for businesses to take into account when developing marketing and product development plans. Now much of the information is publicly available on the Web. As a result, the number of reviews that a product receives grows rapidly. Some popular products can get thousands of reviews or more at some large merchant sites. Many reviews are also long, which makes it hard for potential customers to read them to make an informed decision on whether to purchase the product. A large number of reviews for a single product may also make it harder for individuals to evaluate the true underlying quality of a product. In these cases, customers may naturally gravitate to reading a *few reviews* in order to form a decision regarding the product and he/she only gets a biased view of the product. Similarly, manufacturers want to read the reviews to identify what elements of a product affect sales most. And, the large number of reviews makes it hard for product manufacturers or business to keep track of customer's opinions and sentiments on their products and services.

Recent work has shown that the distribution of an overwhelming majority of reviews posted in online markets is bimodal [8]. Most of the works are based on the assumptions that the rating provided is either binary – good or bad, or star rating on a scale of 1 to 5. In such types of rating we lose the polarity because the meaning is very vague and does not give a clear picture of the product. For example a 3 star rating for a product does not give the indications that which features of the product are not up to the mark when the website allots a rating below the maximum permissible range. In such situations, the average numerical star rating assigned to a product may not convey a lot of information to a prospective buyer. Instead, the reader has to read the actual reviews to examine which of the positive and which of the negative aspect of the product are of interest. Several sentiment analysis approaches have proposed to tackle this challenge up to some extent. However, most of the classical sentiment analysis mapping the customer reviews into binary classes – positive or negative, fails to identify the product features liked or disliked by the customers especially for sentences where the user is not writing complete sentences but express their opinion in two or three words. For example when an opinion is expressed as “Great zoom” or “Excellent picture quality” the parser will not parse it correctly as per the rules designed to extract features as the sentence is not fully correct linguistically and semantically. The rule based design as proposed by few researchers will not pick these sentences from the corpus thereby decreasing the recall value of the overall system.

In this paper, we present an opinion mining system which uses linguistic and semantic analysis of text to identify key information components from text documents from long and short sentences. The short sentences are taken care by identifying the length of sentences and analyzing the parsed text using the Stanford Parser and identifying that the sentence falls in the category of short sentences. The information components are centered on both product features, and associated opinions, which are extracted using natural language processing techniques and co-occurrence-based analysis. Since only those features on which customers express their opinions are of interest, we define an information component as a triplet <F, M, O> where, *F* and *O* represents product feature and opinion respectively. M is an optional component representing adverbs that act as modifier and used to intensify the opinion O. M is also used to capture the negative opinions explicitly expressed in the review. The novelty of the system lies in mining associated modifiers with opinions from short sentences containing two or three words and also features which are expressed as double words like *picture mode, picture quality, touch screen, manual mode* etc . For example, consider following snippets of opinion sentences: (i) *the zoom is very good*; (ii) *the zoom is almost good*. In both of the sentences the opinion word is *good* but the associated modifiers are different that express the degree of customer satisfaction on *zoom*. Another example is a sentence like *I like the large touch screen of the camera,* in which the feature is specified as a double word *touch screen* which has to be taken together in order to classify it as a feature otherwise if we take touch and screen separately then the concept will be wrong as touch is not a feature of the camera . For each extracted feature, the list of opinions and associated modifiers are compiled and their polarity is established using numerical scores obtained through Senti-WordNet. We also present a table that provides the recall value of the features that were obtained by the system.

The remaining paper is structured as follows. Section 2 presents a brief introduction to related work. Section 3 presents the architectural details of proposed opinion mining system. The experimental setup and evaluation results are presented in section 4. Finally, section 5 concludes the paper with possible enhancements to the proposed system.

II. RELATED WORK

Opinion Mining is the task of extracting the opinion expressed by the *source* on some *target* in a given set of document. In this paper *opinion mining* appears as a process of identifying and extracting a list of product features, and aggregating opinions about each of them from review documents. Research on opinion mining started with identifying opinion bearing words, e.g., *great*, *amazing*, *wonderful*, *bad*, *poor* etc. Many researchers have worked on mining such words and identifying their semantic orientations. In [3,4], a bootstrapping approach is proposed, which uses a small set of given seed opinion words to find their synonyms and antonyms in WordNet. The history of the phrase *sentiment analysis* parallels that of *opinion mining* in certain respects. A sizeable number of papers mentioning *sentiment analysis* focus on the specific application of classifying customer reviews as to their polarity – *positive* or *negative* [5,7]. Given a set of evaluative documents D, it determines whether each document $d \in$ D expresses a positive or negative opinion (or sentiment) on an object. For example, given a set of movie reviews, the system classifies them into positive reviews and negative reviews. This classification is said to be at the document level as it treats each document as the basic information unit. Apart from the document-level sentiment classification, researchers have also studied classification at the sentence-level, i.e., classifying each sentence as a subjective or objective sentence and/or as expressing a positive or negative opinion [4].

Although, classical sentiment classification attempts to assign the review documents either positive or negative class, it fails to find what the reviewer or opinion holder likes or dislikes. A positive document on an object does not mean that the opinion holder has positive opinions on all aspects or features of the object. Likewise, a negative document does not mean that the opinion holder dislikes everything about the object. In an evaluative document (e.g., *a product review*), the opinion holder typically writes both positive and negative aspects of the object, although the general sentiment on the object may be positive or negative. Many times the opinion holder writes the opinion without using proper English sentences but the inner meaning of the opinion is quite clear because he/she writes the adjective or adverb followed by the features which are normally nouns. To obtain detailed aspects, feature-based opinion mining is proposed in literature [1,3,6,12]. In [1], a supervised pattern mining method is proposed. In [3, 6], an unsupervised method is used. A lexicon-based approach has been shown to perform quite well in [2, 3]. The lexicon-based approach basically uses opinion words and phrases in a sentence to determine the orientation of an opinion on a feature.

III. PROPOSED FEATURE AND OPINION MINING SYSTEM

Fig.1 presents the architectural details of the proposed feature and opinion mining system, which consists of six major modules – *Document Processor, Subjectivity/ Objectivity Analyzer, Document Parser, Classifier, Feature and Opinion Learner, and Result Analyzer.*

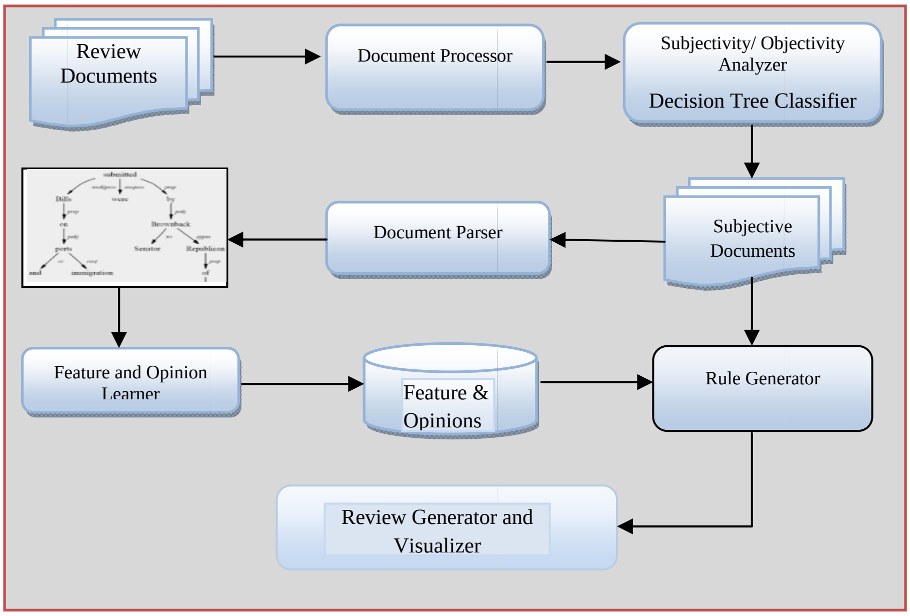


Fig. 1. Architecture of the proposed opinion mining system

A. DOCUMENT PROCESSOR

The *Document Processor* consists of a Markup Language (ML) tag filter, divides the unstructured web documents into individual record size chunks, cleans them by removing the ML tags, and presents them as individual unstructured record documents for further processing by the second module. This module does not in any way alter the contents of the document. It neither cleans it by removing any text data or introducing any missing words or text.

The cleaned document are converted into numeric-vectors using unigram model for the purpose of subjectivity/objectivity analysis.

B. SUBJECTIVE/OBJECTIVE ANALYZER

Pang and Lee [7] has said that Subjective sentences are expressive of the reviewer's sentiment about the product, and objective sentences expresses fact and do not have any direct or obvious bearing on or support of that sentiment. In the document vector given as an output from the *Document Processor* a numeric value represents the likelihood of each word being in a subjective or objective sentence. Therefore, the idea of subjectivity analysis is used to retain segments (sentences) of a review that are more subjective in nature and filter out those that are more objective. This increases the system performance both in terms of *efficiency* and *accuracy*. The idea proposed by Yeh [9] is used to divide the reviews into subjective parts and objective parts. In [9], the idea of cohesiveness is used to indicate segments of a review that are more subjective in nature versus those that are more objective. We have used a corpus of subjective and objective sentences as described in [7] for training purpose. The training set is used to get the probability of each word to be subjective or objective. The decision tree classifier of Weka[1] is trained to classify the unseen review documents into one of the subjective and objective classes. The sentence falling in the objective class are discarded as they are of little importance for opinion mining.

C. DOCUMENT PARSER

Since our aim is to extract product features and the opinions from text documents, all subjective sentences are parsed using Stanford Parser[1], which assigns Parts-Of-Speech (POS) tags to English words based on the context in which they appear. The POS information is used to locate different types of information of interest inside the text documents. For example, generally noun phrases correspond to product features, adjectives represent opinions, and adverbs are used as modifiers to represent the degree of opinion expressiveness. Since, it is observed that opinion words and product features are not independent of each other rather directly or indirectly inter-related through some semantic relations, each sentence is converted into dependency tree using Stanford Parser. The dependency tree, also known as word-word relationship, encodes the grammatical relations between every pair of words. A sample POS tagged sentence and the corresponding dependency tree generated using Stanford Parser is shown in figure 2(a) and 2(b) respectively.

D. FEATURE AND OPINION LEARNER

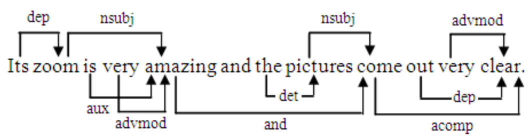
This module is responsible to extract feasible information component from review documents. Later, information components are processed to identify product features and opinions. It takes the *dependency tree* generated by *Document Parser* as input and output the feasible information component after analyzing noun phrases and the associated adjectives possibly preceded with adverbs. On observation, we found that product features are generally noun phrases and opinions are either only adjectives or adjectives preceded by adverbs. For example, consider the following review sentence:

- (ROOT(S(NP(NP (DT The) (NN battery) (NN life))(PP (IN of) (NP (NNP Nokia) (NNP N95))))(VP (VBZ is)(ADJP (RB very) (JJ good)))(. .)))

In the above sentence, “battery life” is a noun phrase and appears as one of the features of Nokia N95 whereas, the adjective word “good” along with the adverb “very” is an opinion to express the concern of reviewer. Therefore, we have defined the information component as a triplet <F, M, O> where, F is a noun phrase and *O* is adjective word possibly representing product feature. M represents adverb that act as modifier and used to intensify the opinion O. M is also used to capture the negative opinions explicitly expressed in the review.

Its/PRP$ zoom/NN is/VBZ very/RB amazing/JJ and/CC the/DT pictures/NNS come/VBP out/IN very/RB clear/JJ ./.

(a) A POS-tagged sentence



(b) Dependency tree

<zoom, very, amazing> // Extracted information component through Rule-1

<pictures, very, clear> // Extracted information component through Rule-2

(c) Extracted Information Components

Fig. 2. (a) A POS-tagged sentence, **(b)** the corresponding dependency tree generated by Stanford Parser, and **(c)** extracted information components

INFORMATION COMPONENT EXTRACTION

---

[1] http://nlp.stanford.edu/software/lex-parser.shtml

The information component extraction mechanism is implemented as a rule-based system which analyzes dependency tree to extract information components. Some sample rules are presented below to highlight the function of the system.

Rule 1: In a dependency tree T , if there exists a *subj*(w_i, w_j) relation such that POS(w_i) = JJ*, POS(w_j) = NN*, w_i and w_j are not stop-words[2] then w_j is assumed to be a *feature* and w_i as an *opinion*. Thereafter, the relation *advmod(*w_i, w_k*)* relating w_i with some adverbial words w_k is searched. In case of the presence of *advmod* relation, the information component identified as <w_j, w_k, w_i> otherwise <w_j, -, w_i>.

Rule 2: In a dependency tree T , if there exists a *subj*(w_i, w_j) relation such that POS(w_i) = VB*, POS(w_j) = NN*, and w_j is not a stop-word then we search for acomp(w_i, w_m) relation. If acomp relation exists such that POS(w_m) = JJ* and w_m is not a stop-word then w_j is assumed to be a *feature* and w_m as an *opinion*. Thereafter, the modifier is searched and information component is generated in the same way as in rule 1.

Rule 3: In a dependency tree T if there exists amod(w_i, w_j) such that POS(w_i) = NN*, and POS(w_j) = JJ* then w_i is assumed to be a *feature* and w_j as an *opinion*. Further if advmod(w_i, w_k) exists then w_k is assumed to be a modifier.

Rule 4: In a dependency tree T if there exists NN(w_i, w_j) such that POS(w_i) = NN*, and POS(w_j) = NN* and Nsubj(w_k, w_i) exists where POS(w_k) = JJ* or amod(w_i, w_k) exists where POS(w_k) = JJ* then w_j,w_i (two words) is assumed to be a *feature* and w_k as an *opinion.*

Figure 2(c) presents two sample information components extracted by applying these rules on the dependency tree shown in figure 2(b). The algorithm, shown in table 1, presents the implementation details of this system.

Table 1: Information component extraction algorithm

Algorithm: Information_Component_Extraction($\mathcal{F}_T$)

```
Input: F_T - a forest of dependency trees
Output: L_IC - information components
L_IC ← φ
for each T ∈ F_T do
If ∃ at least one relation subj(w_i, w_j) in T then
for each relation subj(w_i, w_j) ∈ T do
feature ← opinion ← modifier ← " " // null string
if POS(w_j) = NN* && w_j ∉ L_SW then // L_SW is a list of stop words
if POS(w_i) = JJ* then feature ← w_j ; opinion ← w_i
if ∃ advmod(w_j, w_m) ∈ T then modifier ← w_m
end if
L_IC ← L_IC ∪ {<feature, modifier, opinion>}
else if POS (w_i) = VB* then
if ∃ a relation acomp(w_i, w_k) ∈ T then feature ← w_j ; opinion ← w_k
if ∃ advmod(w_j, w_m) ∈ T then modifier ← w_m
end if
L_IC ← L_IC ∪ {<feature, modifier, opinion>}
end if
end if
end if
end if
end for
else for each amod(w_i, w_j) ∈ T such that POS(w_i) = NN* && w_i ∉ L_SW do
if ∃ a relation amod(w_i, w_k) or nn(w_i, w_k) ∈ T then
if POS(w_k)= VBG then feature ← w_k + w_i ; opinion ← w_j
else if POS(w_j)=RB* then feature←w_i ; opinion←w_k; modifier←w_j
else feature ← w_i ; opinion ← w_j; modifier ← ""
end if
else feature ← w_i ; opinion ← w_j; modifier ← ""
if ∃ advmod(w_j, w_m) ∈ T then modifier ← w_m
end if
end if
end if
L_IC ← L_IC ∪ {<feature, modifier, opinion>}
If ∃ at least one relation amod(w_i, w_j) in T then
for each relation amod(w_i, w_j) ∈ T do
if POS(w_i)= NN* && POS(w_j)= JJ* then
```

[2] A list of 571 stop-words available at http://www.aifb.uni-karlsruhe.de/WBS/aho/clustering

```
feature ← wi ; opinion ← wj
end if
If ∃at least one relation advmod(wi, wk) in T then
modifier ← wk
endif
end for
end if
If ∃at least one relation NN(wi, wj) in T then
for each relation NN(wi, wj) ∈ T do
if POS(wi)= NN* && POS(wj)= NN* then
If ∃at least one relation Nsubj(wk, wi) in T then
for each relation Nsubj(wk, wi) ∈ T do
if POS(wk)=JJ* || amod(wi, wk)in T then
feature ← wj, wi ; opinion ← wk
end if
end for
end if
end if
end if
end for
end if
end for
end if
end for
return LIC
```

E. FEATURE AND OPINION EXTRACTION

Though a large number of commonly occurring noun and adjective phrases are eliminated due to the design of the information component itself, it is found that further processing is necessary to consolidate the final list of information components and thereby the product features and opinions. During the consolidation process, we take care of two things. In the first stage, since product features are the key noun phrases on which opinions are applied, so a feasible collection of product features is identified using term frequency (tf) and inverse document frequency (idf). In the second stage of analysis, however, for each product feature the list of all opinions and modifiers is compiled that are used later for polarity determination of the opinion sentences.

The tf-idf value for each noun phrase is calculated using equations 1 and 2 where, $tf(t_i)$ is the number of documents containing t_i, $|D|$ is the total number of documents and $\left|\left\{ d_j : t_i \in d_j \right\}\right|$ is the number of documents where t_i appears. All those noun phrases having tf-idf value above a threshold are considered as relevant features. Thereafter, for each retained feature, the list of opinion words and modifiers are compiled from information components and are stored in a structured form.

$$tf\text{-}idf(t_i) = tf(t_i) \times idf(t_i) \qquad (1)$$

$$idf(t_i) = \log\left(\frac{|D|}{\left|\left\{ d_j : t_i \in d_j \right\}\right|}\right) \qquad (2)$$

A partial list of product features, opinions, and modifiers extracted from a corpus[3] of 286 customer reviews on *digital camera* shown in table 2.

[3] Review documents were downloaded from http://catalog.ebay.com/

Table 2: A partial list of extracted features, opinions and modifiers

Product	Feature	Modifier	Opinion
Digital Camera	picture	not, really, very	beautiful, clear, fantastic, good, great, professional, sharp
	battery	Very	decent, excellent, rechargeable
	Price	---	cheap, excellent, good, great, high

F. REVIEW GENERATOR AND VISUALIZER

This module exploits the extracted features and opinions to analyze the polarity of opinion sentences in review documents and generates a summarized view. The feature-based review summary can help the customers as well as manufacturers to know about the positive and negative aspects of the products without going through pile of documents. The working principle of this module can be summarized as follows:

- Firstly, the polarity of extracted opinions for each feature are classified using Senti-WordNet [11], a lexical resource in which each WordNet synset s is associated to three numerical scores Obj(s), Pos(s) and Neg(s), describing how objective, positive, and negative the terms contained in the synset are. A partial list of opinions and their positive polarity values (shown in parenthesis) obtained through Senti-WordNet is beautiful (0.75), clear (0.5), fantastic (0.875), good (0.75), great (0.625).
- For each feature, the opinion sentences are examined and mapped into one of the *positive* or *negative* class based on the score value of the associated opinions obtained in the previous step. The *objective* class is not considered as most of the users are interested in either *positive* or *negative* views rather than *neutral* views. The *max* function is applied to decide the class of an opinion sentence. In case of presence of multiple features in an opinion sentence, the one having highest score value is used to decide its class.
- Finally, we calculate the precision and recall after applying the rules on individual models of camera and iPhone to generate a feature based summary.

IV. EXPERIMENTAL RESULTS

In this section, we present the experimental details of the proposed opinion mining system. For subjectivity analysis, we used the subjectivity dataset[4] v1.0 from Cornell for training purpose. The dataset consists of 1000 subjective sentences and 1000 objective sentences. A Java program is written to extract features using unigram model from this dataset and to convert each sentence into equivalent numeric vector where a value represents likelihood of each word being in a subjective or objective sentence. Thereafter, the Decision Tree classifier of Weka is trained to classify the unseen sentences into subjective and objective classes. The accuracy of the classier on 10-fold cross validation is 82%. The data sample used in our work to mine features and opinions for customer reviews summarization consists of 286 review documents on different models of digital camera (Canon: 60, Kodak: 100, Nikon: 100, Panasonic: 26) and 50 documents on iPhone – all obtained from www.ebay.com. The algorithm presented in table 1 was implemented using Java to mine features and opinionated words along with modifiers from the subjective review sentences. Initially, a total of 131 and 33 features for *digital camera* and *iPhone* respectively were extracted out of which only 20 and 14 were retained after feasibility analysis. For each retained feature, the list of both opinions and modifiers were compiled, a partial view of which has been already shown in table 2. Thereafter, the sentiment polarity of all opinions were obtained Senti-WordNet to generate the feature-based positive and negative review summaries as discussed in section 3.3

A. EVALUATION METHODS

We now present a discussion on the performance of the whole system which is analyzed by taking into account the performance of the *feature and opinion* extraction process. Since terminology and complex proper names are not found in Dictionaries, an obvious problem of any automatic method for concept extraction is to provide objective performance evaluation. Therefore manual evaluation has been performed to judge the overall performance of the system. For evaluation of the experimental results, we use standard IR performance

[4] http://www.cs.cornell.edu/people/pabo/movie-review-data/

measures. From the extraction results, we calculate the true positive *TP* (number of correct feature-opinion pairs the system identifies as correct), the false positive *FP* (number of incorrect feature-opinion pairs the system falsely identifies as correct), true negative *TN* (number of incorrect feature-opinion pairs the system identifies as incorrect), and the false negatives *FN* (number of correct feature-opinion pairs the system fails to identify as correct). By using these values we calculate the following performance measures:

- **Precision (π):** the ratio of true positives among all retrieved instances.

$$\pi = \frac{TP}{TP + FP} \qquad (3)$$

- **Recall (ρ):** the ratio of true positives among all positive instances.

$$\rho = \frac{TP}{TP + FN} \qquad (4)$$

- **F1-measure (F_1):** the harmonic mean of recall and precision.

$$F_1 = \frac{2\rho\pi}{\rho + \pi} \qquad (5)$$

- **Accuracy (τ):** the ration of sum of true positives and true negatives over total positive and negative instances.

$$\tau = \frac{TP + TN}{TP + FP + FN + TN} \qquad (6)$$

Table 3. Performance evaluation of the feature-opinion extraction process

Product Name		With Rules 1 & 2 only			With Rules 1, 2, 3 & 4		
		Precision (%)	**Recall (%)**	**F1-measure (%)**	**Precision (%)**	**Recall (%)**	**F1-measure (%)**
Digital Camera	**Canon**	92.50	57.81	71.15	82.45	69.42	75.38
	Kodak	94.83	42.97	59.14	86.12	58.67	69.79
	Nikon	91.67	41.12	56.77	79.20	52.45	63.11
	Panasonic	91.43	64.00	75.29	83.45	69.10	75.60
iPhone		85.19	48.94	62.16	79.25	53.50	63.88
Macro-Average		91.12	50.97	64.90	82.09	60.63	69.55

The values of the above performance measures are calculated for each category of experimental data. In order to present a synthetic measure of performance over all categories, we present the macro-averaged performance which consists in simply averaging the result obtained on each category. Table 3 summarizes the performance measure values for our system in the form of a misclassification matrix. The recall value is lower than precision indicating that certain correct feature-opinion pairs could not be recognized by the system correctly. This is justified since most of the reviewers do not follow the grammatical rules while writing reviews due to which the parser fails to assign correct POS tag and thereby correct dependency relations between word pairs. However, the precision value is high which indicates that the identified feature-concept pairs are correct, which leaves scope for enhancing our grammar to accommodate more dependency relations. The value of precision is lower in the second case when we are using all the four rules which is justified because some of the features extracted by the system are not correct as mining small sentences pose a problem that some sentences do not contain features when extracted by rule 3. For example a sentence like “very good light” or “excellent case” are also extracted but light and case are not the features of the phone. However, if we put a threshold value and filter out those features which are less than the minimum support then these features can be filtered out. The recall value is quite high in the second case which is what we wanted as new sentences were not extracted before using previous two rules were correctly identified by applying Rule 3 and 4. More work is needed to enhance the precision values which will make the system fully automated to extract features with high precision and recall. After analyzing the review documents manually we also found that some review documents contain junk sentences too which opens a new direction of research – review spam analysis.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed the design of an opinion mining system which performs linguistic and semantic analysis of text to identify product features and opinions from customer review documents from long sentences and short sentences. We have also proposed a rule based method to identify the sentiment polarity of opinion sentences for the purpose of feature-based review summarization. Presently, we are refining the rule-set to consider more dependency relations to improve the *precision* values of the system. Moreover, we observed that most of the reviewers use fuzzy terms (linguistic qualifiers) rather than crisp terms while writing a review. Motivated by this observation, we are developing a system which will consider more sentences which were not extracted earlier so that the precision and recall values are further improved. The identification of such a system will pave a way for a fully automatic system which will rank the features and its associated modifiers with a high degree of accuracy.

REFERENCES

[1] B.Liu, M. Hu, and J.Cheng, “Opinion Observer - Analyzing and comparing opinions on the Web”, in *Proceedings of the 14th International Conference on World Wide Web (WWW’05), Japan*, 2005, pp. 342-351.

[2] X.Ding, B.Liu, and S.Y. Philip, “A Holistic Lexicon-Based Approach to Opinion Mining”, in *Proceedings of the first ACM International Conference on Web search and Data Mining (WSDM’08),* California, USA, 2008, pp. 231-240.

[3] M.Hu, and B.Liu, “Mining and Summarizing Customer Reviews”, in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’04), USA,* 2004, pp. 168 – 177.

[4] S.Kim, and E.Hovy, “Determining the Sentiment of Opinions”, in *Proceedings of the 20th International Conference on Computational Linguistics (COLING’04), Switzerland,* 2004, pp. 1367-1373.

[5] B.Pang, L.Lee, and S.Vaithyanathan, “Thumbs up? Sentiment Classification Using Machine Learning Techniques”, in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP’02), USA,* 2002, pp. 79 – 86.

[6] A.M.Popescu, and O. Etzioni: “Extracting Product Features and Opinions from Reviews”, *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP’05)*, *Canada,* 2005, pp. 339 – 346.

[7] P.Turney, “Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews”, in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL’02), Philadelphia, Pennsylvania,* 2002, pp. 417 - 424.

[8] A.Ghose, and P.G.Ipeirotis, “Designing Ranking Systems for Consumer Reviews: The Impact of Review Subjectivity on Product Sales and Review Quality”, in *Proceedings of the Workshop on Information Technology and Systems (WITS’06),* Milwaukee, December 2006.

[9] E.Yeh, “Final Project Picking the Fresh from the Rotten: Quote and Sentiment Extraction from Rotten Tomatoes Movie Reviews”, CS224N/Ling237, 2006.

[10] B.Pang, B. and L.Lee, “A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts”, in *Proceedings of ACL 2004*, 2004, pp. 271-278.

[11] A.Esuli, and F.Sebastiani, “SentiWordNet: A publicly available lexical resource for opinion mining”, in *Proceedings of LREC-06, the 5th Conference on Language Resources and Evaluation*, Genova, IT, 2006, pp. 417-422.

[12] M.Abulaish, Jahiruddin, M.N.Doja, T.Ahmad, “Feature and Opinion Mining from Customer Review Documents”, in Proceedings of Pattern Recognition and Machine Intelligence, 2009, (PReMI 2009),pp.: 219-224