

# Web User Opinion Analysis for Product Features Extraction and Opinion Summarization

Ravi kumar V<sup>1</sup> and K. Raghuv<sup>2</sup>

<sup>1</sup>Research Scholar, Department of Information Science &Engineering, NIE,Mysore,  
Karnataka, India

ravikumarv@nie.ac.in

<sup>2</sup> Faculty, Department of Information Science &Engineering, NIE,Mysore, Karnataka,  
India

raghuniel1967@gmail.com

## **ABSTRACT**

*Selling the product through Web has become more popular because of online shopping. This enables merchants to sell their products through Web and expects the customer to express their opinion through online about the product which they have purchased. Due to this we find number of customer reviews on a particular product, it varies from hundreds to thousands, for some product it is more than that. In order to help the customer and the manufacture/merchant we propose a semantic based approach to mine different product features and to find the opinion summarization about each of these extracted product features by means of web user opinion expressed through the customer reviews using typed dependency relations.*

## **KEYWORDS**

*Product Feature extraction, Opinion Identification, Opinion Sentences, Dependency Relations, Pronoun resolution.*

## **1. INTRODUCTION**

Many times the customers purchase a product and express their experience in their reviews. The reviews have been written by different customers, written spontaneously expressing their opinion about different features of a product in the form of a short paragraph not pertaining to a particular topic, but it conveys the sentiment about the product which they have purchased. This type of text documents have become more popular and challenging research work to explore and to extract useful information. As we get thousands of customer reviews about a product from different sources in the web like E-commerce sites , opinion sites , forums, blogs, etc. it becomes difficult for a customer to analyze the data to take right decision in purchasing any product. If the information is available in the form of short summary, i.e. it consists of many product features along with the opinion expressed by the different customers on each features, helps the new customer to take right decision.

In the past few years various product feature extraction techniques have been proposed by many researchers [1, 2, 3, 4]. Amongst, the technique of [1], referred as one of the elementary method to extract product features and opinion on these features is fully automated because, it does not involve the use of training examples. However, the design principle of [1] does not anchored in a semantic perspective. As a result, it is ineffective in excluding non-product features and opinion-irrelevant product features. To deal with these restrictions, we propose a semantic-based product feature extraction technique. Each word in a sentence will have some grammatical relations between each other based on their Part of Speech (POS) definitions, positions in the sentence etc. Using the Stanford typed dependency relations [5] between each word present in the sentence and the opinion lexicon consisting of a list of subjective positive and negative words from subjectivity clue (<http://www.cs.pitt.edu/mpqa/lexiconrelease>), General Inquire (GI) DOI : 10.5121/ijwest.2012.3407

(<http://www.wjh.harvard.edu/inquirer/>), we identify the product features which occurs frequently and as well as infrequently from the given customer review. Here an attempt has been made to extract product feature using the semantic relation between each word present in the sentence.

Many natural language processing applications such as question answering, information extraction, and so on requires subjectivity classification to classify the sentences or the documents which contain opinions from factual [6]. Opinion sentence identification has been carried out by many researchers by means of determining the presence of specific parts-of-speech such as adjectives, adverbs, etc. or a list of seed words that may potentially represent opinions [7]. Opinion summarization is the process of producing a sentiment summary [1, 3, 8, 9], consists of sentences from reviews that capture the customers opinion on product features or objects on which customers have expressed the opinions. Here we have enhanced the approach of [1], to identify opinion sentences from the customer review by replacing the pronoun present in the review sentence with appropriate product feature and by considering opinion words as adjectives, verbs and even nouns are also as opinion words. Finally we generate summary consisting of positive and negative opinion sentences attached to each product features expressed by different customers from their reviews.

## **2. RELATED WORK**

Minqing Hu et al. considered as the pioneer work to find the summarization based on feature and opinion. They have used association rule mining to find frequent item sets, obtained from each sentence noun phrases. To prune the frequent items they have used different techniques. The infrequent features are identified based on the opinion word present in the sentence. Summary consisting of the product feature and the opinion about it has been given in terms of positive and negative [1].

Gamgarn Somprasertsri et al. proposed an approach for mining product feature and opinion based on the consideration of syntactic and semantic information. They have used dependency relations and ontological knowledge with probabilistic model. They have also used Product Ontology to identify similar feature with different terminology [10].

YuanbinWu et al constructed their own dependency parser, to identify the product features and the opinion on these features from the product reviews. The opinion associated with the feature is identified based on the window size of 5 from the extracted word to the opinion word present in that sentence [11].

Parma Nand proposed an algorithm for resolving anaphora based exclusively on salience weights. He has focused on resolving anaphors particularly in the genre of in short newspaper type articles because it forms part of wider research aimed at building a system for visualization of online newspaper articles. The developed algorithm is capable of resolving the anaphors using knowledge-poor approach which is completely based on salience scores. It works on semi parsed text and currently only uses syntax and surface level semantics of the language to resolve primarily pronominal pronouns [12].

Chih-Ping Wei et al. used the approach [1], to mine product features and opinion about these features using the semantic based approach. Once again the approach is based on co occurrence of noun phrase and the opinion word [13].

Wilson et al. used syntactic clues derived from dependency parse tree as features for predicting the intensity of opinion phrases [14].

j. Wiebe Opinion sentence identification has been carried out by determining the presence of specific parts-of-speech such as adjectives, adverbs, etc. or a list of seed words that may potentially represent opinions[7].

Ashequl Qadir used typed dependency relations between each word in a sentence to identify opinion sentence. It is going to assign a product feature to each of the opinion sentences based on the frequently associated words that appear with in the selected typed dependency relations [15]. Hatzivassioglou et al. showed that adjectives can potentially contribute toward identifying subjective sentences [16].

### 3. PROPOSED APPROACH

The proposed approach has got two main subtasks. The first task is to extract various product features of a given product. The second task is to recognize the opinion sentences specific to the extracted product features and to find summary based on the users opinion.

The architecture of the proposed approach for the first task is shown in Fig.1 and the steps involved in this process are explained in the following section in detail.

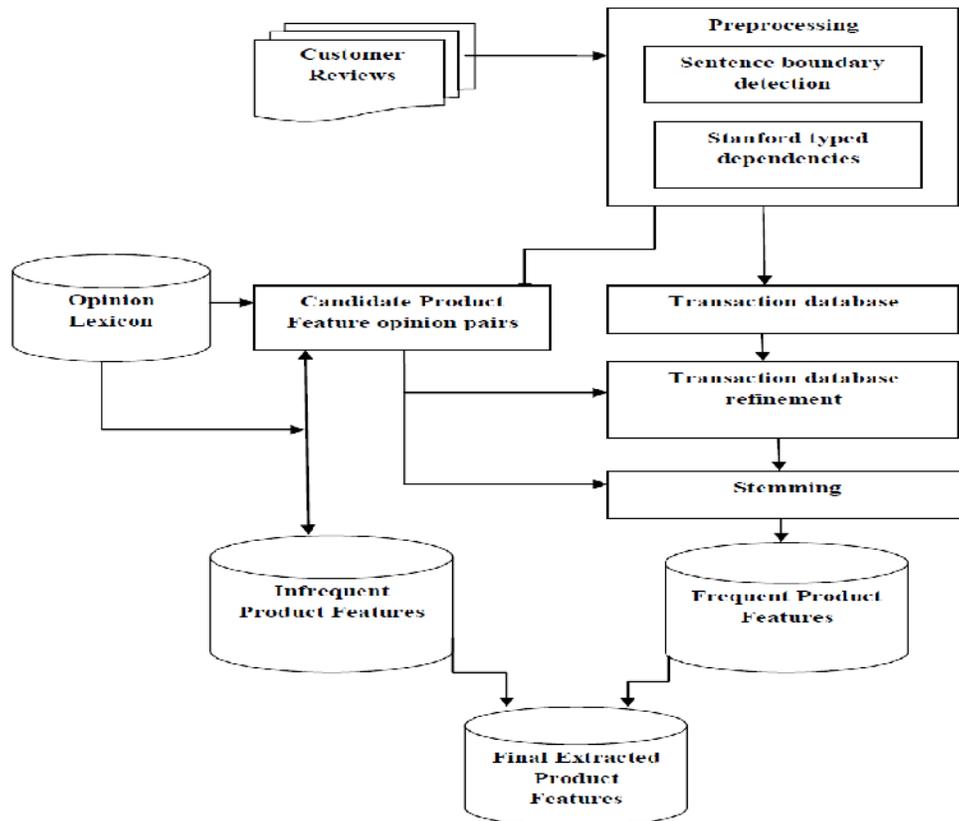


Fig.1: Architecture to extract final product features

#### 3.1. Preprocessing

The preprocessing involves identification of sentence boundary, because the Stanford parser expects a sentence as an input to generate typed dependencies and the Collapsed Dependencies

for the given customer review and this can be achieved using Ling Pipe Sentence Boundary method. The output from the parser for the sentence 'The battery is good', looks like:

(ROOT [31.403]

(S [31.297]

(NP [13.753] (DT The) (NN battery))

(VP [10.320] (VBZ is)

(ADJP [5.450] (JJ good)))

(. . .))

[det(battery-3, The-2), nsubj(good-5, battery-3), cop(good-5, is-4)]

Where 'det', 'nsubj' and 'cop' are represents the dependency relation between each of word present in the given sentence.

### 3.2. Extracting candidate product feature opinion pairs

The candidate product feature opinion pairs are extracted in an effective and meaning full way using different combinations of dependencies. The noun or noun phrase (candidate product feature) extracted is present either before or after the verb phrase in the parse tree. This indicates that the candidate product feature may be present either in the subject part or in the object part of a given sentence. In our approach we have considered opinion word/s associated with the candidate product feature/s may be an adjective/s or a verb/s or a noun/s.

1) nsubj - Nominal Subject: A nominal subject is a noun phrase which is the syntactic subject of a clause. The governor of this relation might not always be a verb: when the verb is a copular verb, the root of the clause is the complement of the copular verb [5].

nn - Noun Compound Modifier: A noun compound modifier of an NP is any noun that serves to modify the head noun[5]. Suppose the product feature is of more than one word then we can find this by using this dependency.

Example: "The battery life is good".

nsubj (good-6, life-4) nn(life-4

The extracted candidate product feature, opinion pair is: (battery life, good)

2) If the second argument to the dependency "nsubj", is not a noun but the first argument is an adjective, then we need to identify the noun associated with this adjective. This is the case when the feature is present as an object in the sentence.

xcomp- Open Clausal Complement: An open clausal complement (xcomp) of a VP or an ADJP is a clausal complement without its own subject, whose reference is determined by an external subject [5].

dobj- Direct Object: The direct object of a VP is the noun phrase which is the (accusative) object of the verb[5].

Using these three dependencies we can find the feature and the opinion associated with it.

Example: It is great having the LCD display.

nsubj(great-4, It-2) xcomp(great-4, having-5)

dobj(having-5, display-8) nn(display-8, LCD-7)

The extracted candidate product feature, opinion pair is: (LCD display, great).

3) When the verb is considered as the opinion word and the feature is present in the object part of the sentence we have used 'nsubj' and 'dobj' dependencies to identify feature, opinion pair.

Example: I like this camera.

nsubj (like-3, I-2) dobj (like-3, camera-5)

The extracted candidate product feature, opinion pair is: (camera, like).

4) acomp - Adjectival Complement : An adjectival complement of a verb is an adjectival phrase which functions as the complement (like an object of the verb) [5]. We use this dependency along with the 'nsubj' when the opinion word is a verb and adjective becomes the complement of this verb.

Example: The optical zoom works great.

nsubj(works-5, zoom-4) acomp(works-5, great-6)

The extracted candidate product feature, opinion pair is: (zoom, great).

5) rcomod- Relative Clause Modifier: A relative clause modifier of an NP is a relative clause modifying the NP. The relation points from the head noun of the NP to the head of the relative clause, normally a verb[5]. We use this dependency when the noun part of the dependency 'nsubj' is not a noun and this noun can be found through the verb that connects to 'rcomod'.

Example: It has movie mode that works good for a digital camera.

nsubj(works-7, that-6) acomp(works-7, good-8)  
rcomod(mode-5, works-7) nn(mode-5, movie-4)

The extracted candidate product feature, opinion pair is: (movie mode, good).

6) amod- Adjectival Modifier: An adjectival modifier of an NP is any adjectival phrase that serves to modify the meaning of the NP [8].It has been absorbed that most of the time a feature is modified by an adjective. Hence we have considered this dependency.

Example; "There is a great camera"

amod(camera-6, great-5)

The extracted product feature, opinion pair is: (camera, great).

7) Some time we find that a single feature is associated with more than one opinion words and it becomes necessary to consider all of them to find summary about each feature. This can happen in two situations they are:

i) Consider the dependency 'amod' having the first argument as noun and the second argument as adjective and from this find more dependencies of type 'amod' that has a link with this through adjective.

Example: Nokia has fine, excellent, cheapest battery.

amod(battery-9, cheapest-8)

The extracted product feature, opinion pair is: (battery, cheapest).

amod(battery-9, cheapest-8) amod(cheapest-8, fine-4)

The extracted product feature, opinion pair is: (battery, fine).

amod(battery-9, cheapest-8) amod(cheapest-8, excellent-6)

The extracted candidate product feature, opinion pair is: (battery, excellent).

ii) To find more opinion words associated with a feature we use the dependency 'conj\_and', the relation between two elements connected by a coordinating conjunction, such as "and", "or", etc.

Example: Nokia has fine, excellent and cheapest battery.

amod(battery-9, fine-4)

The extracted candidate feature, opinion pair is: (battery, fine).

amod(battery-9, fine-4) conj\_and(fine-4, cheapest-8)

The extracted candidate feature, opinion pair is: (battery, cheapest).

amod(battery-9, fine-4) conj\_and (fine-4, excellent-6)

The extracted candidate product feature, opinion pair is: (battery, excellent).

8) We can use 'conj\_and', to find an opinion word associated with more than one product.

Example: Nokia has good screen and battery.

amod(screen-5, good-4)

The extracted candidate product feature, opinion pair is: (screen, good).

amod(screen-5, good-4) conj\_and(screen-5, battery-7)

The extracted candidate product feature, opinion pair is: (battery, good).

9) prep\_with- Is the collapsed representation dependencies involving prepositions, as well as information about the referent of relative clauses are collapsed to get direct dependencies between content words [5]. Feature opinion pair can be found using prep\_with dependency along with 'nsubj'.

Example: I am happy with my Nokia phone.

nsubj(happy-4, I-2) prep\_with(happy-4, phone-8) nn(phone-8, Nokia-7)

The extracted product feature, opinion pair is: (Nokia phone, happy).

10) It so happens that some time noun itself express the opinion about other noun. Consider the following examples:

Example: Battery life was never a problem for me.

nsubj (problem-6, life-2) nn (battery-1, life-2)

The extracted candidate product feature, opinion pair is: (problem, battery life).

11) Some time we find customer review sentences consisting of two or more features shares same opinion word.

For example 'This camera will give you a great picture quality, LCD screen, and price.'

amod(quality-10, great-8) conj\_and(quality-10, screen-13)

conj\_and(quality-10, price-16) nn(quality-10, picture-9) nn(screen-13, LCD-12)

The extracted candidate product feature opinion pairs are (picture quality, great), (LCD screen, great) and (price, great).

**Opinion Lexicon:** As we mentioned earlier a noun or noun phrase is said to be a candidate product feature if it is associated or modified by one or more opinion words, it may be an adjectives or a verbs or a nouns. This leads to construction of seed list consisting of positive and negative opinion words. We have considered 2173 positive, 3964 negative words from subjectivity clue <http://www.cs.pitt.edu/mpqa/lexiconrelease> and 773 positive and 803 negative words from General Inquire(GI) (<http://www.wjh.harvard.edu/inquirer/>). From this list we have removed some of the words like 'just, able, common, want etc.' which are not expressing the opinion and they might express the fact.

The algorithm to find the candidate product feature opinion pairs and transaction database is shown in Fig. 2.

### 1. Algorithm Feature\_Opinion\_Pair()

```

//Input:      PS – Parsed sentence
              PSL – Positive Seed List
              NSL-Negative Seed List

//Output:
              F1 – File consisting of transaction data base
              F2 – File consisting of candidate product feature opinion pairs
2. for each dependency di ∈ PS do
3.   if di = 'nsubj' then
4.     if arg1 ∈ 'nsubj' = adjective then
5.       PairExtract_adjective (di, PS, F1, F2, PSL,NSL);
6.     else if arg1 ∈ 'nsubj' = verb then
7.       PairExtract_verb(di, PS, F1, F2,PSL,NSL);
8.     else if arg1 ∈ 'nsubj' = 'n' then
9.       PairExtract_noun(di, PS, F1, F2,PSL,NSL);
10.    endif
11.  else if di = 'amod' then
12.    if arg1 ∈ amod is an adjective then
13.      PairExtract_amod(di, PS, F1, F2,PSL,NSL);
14.    endif
15.  endif
16. endfor
    
```

Fig.2: Algorithm to find feature opinion pairs and transaction database.

Each parsed sentence is processed to find the presence of Noun/Noun phrase and associated opinion word/s using the above mentioned different possible dependency relations. The identified Nouns/Noun phrases are placed in to the transaction data base (F1) and are considered as candidate product feature if and only if the associated opinion word/s either present in the positive or negative list of opinion lexicon mention earlier. At the ends of this process we get Candidate Product Feature Opinion Pair (CPFOP) is a set of elements consisting of Noun/Noun Phrase and opinion word associated and the elements into the transaction database.

**Transaction Database:** It consisting of either a noun or noun phrase obtained during the process of extracting candidate product feature opinion pairs. Each row in a transaction database represents one sentence in the review, consisting of different items (noun/noun phrases) as columns.

This transaction set is different from [1], i.e.in the way that, the items present in each row further refined to get refined transaction database. The rule to consider the element into the refined transaction database is defined as:

$$\begin{aligned}
 I &= \{I_1, I_2, \dots, I_n\} \in T_i \in TD1 \\
 \text{IF } \exists I_i \in T_i \in \{ \text{NP of CPFOP} \} &\text{ then} \\
 T_j &\cup I_i \in TD2
 \end{aligned}$$

Where I is set of elements belongs to transaction T<sub>i</sub> of original transaction database TD1 and if the item I<sub>i</sub> in T<sub>i</sub> is present in the noun part NP of the candidate product feature opinion pair (CPFOP) then add the element into the transaction T<sub>j</sub> of the refined transaction database TD2.

**Stemming:** After finding elements into the transaction database and candidates product feature opinion pairs we apply stemming using Porter Stemmer [17], so that all nouns and noun phrases that are in plural form are converted in to a singular form so that the item set will become stronger.

**Negation identification:** The negation word associated with the opinion word most of the time makes the opinion opposite to the one expressed in the sentence. The possible negation words appeared in the review sentences are ‘not’, ‘never’ and ‘no’. These negations can be identified by using the following dependencies:

**neg-** Negation Modifier: The negation modifier is the relation between a negation word and the word it modifies [8]. This is used when we find ‘not’ and ‘never’ negation words associated with the opinion word.

Example: “The screen is not good”.

nsubj (good-6, screen-3) neg (good-6, not-5)

The extracted feature, opinion pair is: (screen, good).

**det-** determiner: A determiner is the relation between the head of an NP and its determiner [8]. When the opinion word is associated with the negation word ‘no’, we find this negation using the dependency ‘det’.

Example: I have no problem with the battery.

nsubj(have-3, I-2) dobj(have-3, problem-5)

prep\_with(problem-5, battery-8) det(problem-5, no-4)

The extracted feature, opinion pair is: (battery, problem)

### 3.3. Mining frequent product features

The product features are the different attributes of a product and some time the product itself become one of the features. As we know that most of the time a product feature is a noun and it has been modified either by an adjective or verb or a noun itself. As we already extracted nouns/noun phrases from the given customer review for each sentence in the section 3.2. In order to find frequent product features we have used association mining based on the Apriori algorithm [18] on the refined transaction database obtained in the previous step. The reason behind the use of refined transaction database is to avoid pruning process after finding frequent product features and also it reduces the number of transaction so that we get more appropriate frequent product features from association mining. We find frequent features with user defined minimum support of 1%.

### 3.4. Mining Infrequent product features

Most of the existing approaches follow the method of finding noun/noun phrase and the nearest adjective present in the sentence to find infrequent features but are relevant features. This may not be an effective way of finding the infrequent noun/noun phrases, because the adjective extracted may not have any subjective opinion and which leads to the noise in the infrequent feature extraction process. Consider the review sentence: “the picture quality is not rich in color.” The noun closest to the subjective adjective “rich” is “color,” but color is actually not the product feature discussed in the sentence; rather, the true product feature should be “picture quality.” Simply taking into consideration the nearest noun or noun phrase to a subjective adjective, without relating to the order priority or the location of the noun or noun phrase, is not an adequate indication of the exact product feature spoken in the review sentence.

In our approach as we extract the noun/noun phrase and the opinion associated with it using the dependency relation and opinion lexicon respectively (section 3.2). For example for the above mentioned sentence when we apply our approach it extracts the product feature as “picture quality” not as “rich”. With this we include the infrequent product features into the frequent product features FF identified so far using Apriori, i.e.

If  $\{np_i\} \neq \emptyset$  and  $\{ow_i\} \in \{\text{opinion lexicon list}\}$  then  $FPF = FF \cup \{np_i\}$

Where  $\{np_i\}$  set of infrequent product features and  $\{ow_i\}$  opinion word associated with  $np_i$  and present in opinion lexicon list. The final product features are in FPF.

### 3.5. Identification of Opinion Sentences

Definition: Opinion Sentence

A sentence is said to be an opinion sentence if it contains one or more product features and opinion pairs.

We considered each sentence from the review to find Noun/Noun phrases and the associated opinion word/s using the process mentioned in the section 3.2. we consider the pair whose Noun/Noun phrase should present in the extracted product features list and the opinion word should present in the opinion words list (lexicon). During this process of identifying opinion sentence we find existence of pronoun 'it' as a product feature in the product feature opinion pair, resolving this will have an effect on the performance of orientation of opinion sentence. Here we have proposed an algorithm to resolve this pronoun by a proper product feature. The algorithm to resolve pronoun 'it' is shown in Fig.3 below.

#### 1. Algorithm Pronoun\_Rsolution()

**//Input:** S – Sentence to resolve pronoun 'it'  
 PFOP-Product Feature Opinion Pair  
 PS1 and PS2- Two previous sentences to S  
 PN –Product Name  
 FPF- Final Product Features (FPF) extracted using our approach  
 PSL – Positive Seed List  
 NSL-Negative Seed List

**//Output:** S – Sentence after resolving pronoun with proper product feature

2. if (pronoun 'it' present at the beginning of S and S is the first sentence of the review) then
3.     replace pronoun with PN
4. else if (pronoun 'it' present at the beginning of S)then
5.     find the product features present in PS1 and PS2 along with frequency of each product feature in the entire review and consider two topmost product features  $F1, F2 \in FPF$
6.     if ( $\text{freq}(F1) > \text{freq}(F2)$ ) then
7.         replace pronoun with F1
8.     else
9.         replace pronoun with F1
10.         if ( $\text{freq}(F1) = \text{freq}(F2)$ ) then
11.             replace pronoun with the nearest to the 'it' among F1 and F2
12.         endif
13. else if (pronoun 'it' present in between of S) then
14.     find the total number of product features (TPF) present in S and along with frequency of each product feature in the entire review and consider 2 nearest product features  $F1, F2 \in FPF$  if there are more than 2.
15.     if (TPF=1)
16.         replace pronoun with F1

```

17.         else if (freq(F1)>freq(F2) )then
18.             replace pronoun with F1
19.     else
20.         replace pronoun with F1
21.         if (freq(F1)=freq(F2) )then
22.             replace pronoun with the nearest to the ‘it’ among F1and F2
23.         endif
24.     endif
25. endif
26. endif
27. endif
28. endif
29. return S
    
```

Fig.3: The algorithm to resolve pronoun ‘it’

Once the opinion sentence is identified, we place the opinion sentence either into positive or negative category with respect to each product feature based on the opinion word associated with the product feature. During this process of identifying opinion associated with the product feature, we have considered negation associated with the opinion word as explained in the earlier part of the paper. This consideration of negation changes the category of the opinion sentence.

### 3.6. Summary Generation

Now we are in the final process of generating feature based summary of a given product. It is clearly different from traditional text Summarization. We are only interested in features of the product that customer has opinion on and in the sentiment of the customer opinion i.e. is it positive or negative. Summary consisting of each extracted product feature along with the opinion sentences classified into positive and negative category.

The summary generated for the extracted product feature “camera” is as follows

Feature: **Camera**

Positive: **32**

- This camera is perfect for an enthusiastic amateur photographer.
- I enthusiastically recommend this camera.
- Camera is easy to use, and yet very features rich.

....

Negative:1

- Made the camera totally inoperable.

## 4. EXPERIMENTS AND RESULTS

### 4.1 Data Collection

We have conducted experiments using the data set [1], consists of customer reviews of five electronics products: 2 digital cameras, 1 DVD player, 1 mp3 player, and 1 cellular phone collected from Amazon.com. Products in these sites have a large number of reviews. Each of the reviews includes a text review and a title. Each product reviews consists of 100 reviews and to evaluate the discovered product features, a human tagger manually read all the reviews and each opinion sentence is labeled product feature/s and also the opinion associated with each product feature.

### 4.2 Evaluation criteria and comparative evaluation

To evaluate the performance our method to find product features and opinion sentences, we have adopted precision and recall. The definitions of precision and recall are as following.

$$\text{Precision} = \frac{PC}{PM} \quad \text{Recall} = \frac{PC}{PT}$$

Where PC = number of correctly mined product features/opinion sentences.  
 PM = the number of all mined product features/opinion sentences.  
 PT = the number of all correct product features/opinion sentences.

The proposed technique has been evaluated from two perspectives:

- 1) The effectiveness of the product features extraction.
- 2) The effectiveness of the opinion sentence extraction considering all the extracted product features.

**Experiment 1:** Our approach to extract product features on different products has been evaluated using the above mentioned dataset.

As our proposed approach is based on the semantic relation between each word in the sentence and the opinion words list (opinion lexicon) with known semantic orientation, it has the potential to exclude non-product features, opinion-irrelevant product features and efficiently identify product features with infrequent occurrences in the consumer reviews, and eventually improve the effectiveness of product feature extraction. The table1 shows the experimental result.

Table-1: Comparison of precision and recall values of our proposed approach to find different product features with Minqing Hu et al.[1]approach.

	Nikon(%)	Canon(%)	Nokia(%)	Mp3 player(%)	DVD player (%)	Avg (%)
<b>Recall</b>						
HU and Liu	0.822	0.792	0.761	0.818	0.797	0.80
Our Approach	0.835	0.812	0.82	0.824	0.836	0.82
<b>Precision</b>						
HU and Liu	0.747	0.71	0.718	0.692	0.743	0.72
Our Approach	0.733	0.701	0.753	0.707	0.759	0.73

Fig. 4 and Fig. 5, shows the comparison of Recall and Precision values of our proposed approach to find different product features with Minqing Hu et al.[1]

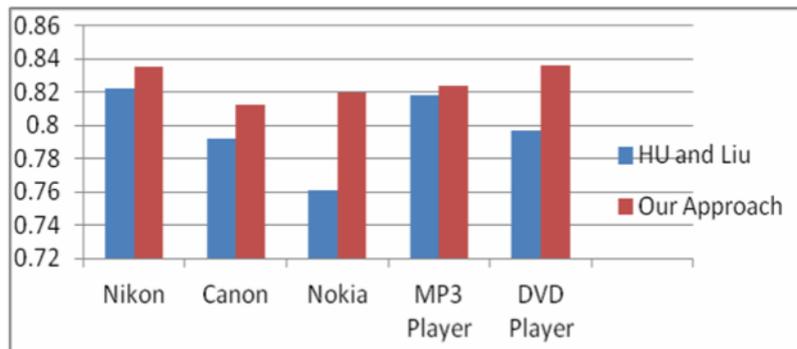


Fig. 4: Comparison of Recall V/s Product, between proposed approach to extract different product features with and Minqing Hu et al.[1]

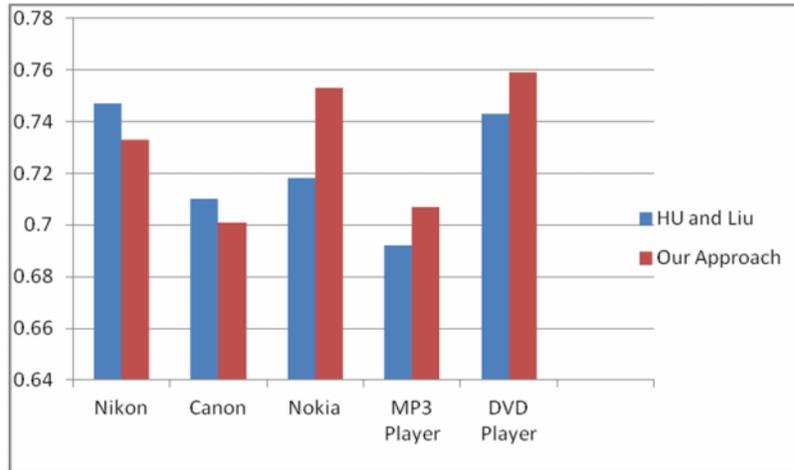


Fig. 5: comparison of Precision v/s Product between proposed approach to extract different product features with and Minqing Hu et al.[1]

**Experiment 2:** The experiment is conducted using data set [1] to find the opinion sentences on different product customer reviews.

Our proposed approach is different from the approach of Minqing Hu et al.[1], in way that, we have not only considered the adjectives as opinion words but along with this we have considered verbs and nouns are also as opinion words. We have also resolved the pronoun ‘it’ by replacing with proper product feature that improves the efficiency of identifying opinion sentence. With these consideration results for opinion sentence identification shows that our approach performs well compared with Minqing Hu et al.[1]. The table2 shows the experimental result.

Table-2: Comparison of precision and recall value result of our approach to identify opinion sentences from the given customer reviews for a range of products with Minqing Hu et al.[1]approach.

	Nikon(%)	Canon(%)	Nokia (%)	Mp3 player(%)	DVD player (%)	Avg(%)
<b>Recall</b>						
HU and Liu	0.719	0.634	0.675	0.784	0.653	0.693
Our Approach	0.751	0.747	0.690	0.685	0.665	0.707
<b>Precision</b>						
HU and Liu	0.643	0.554	0.815	0.589	0.607	0.642
Our Approach	0.711	0.805	0.831	0.646	0.623	0.723

Fig.6 and Fig. 7, shows the comparison of Recall and Precision values of our proposed approach to identify opinion sentences with Minqing Hu et al.[1]

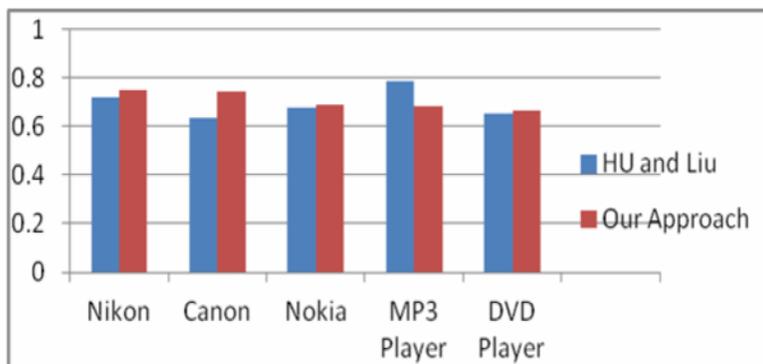


Fig. 6: comparison of Recall V/s Product, between proposed approach to extract opinion sentences with and Minqing Hu et al.[1]

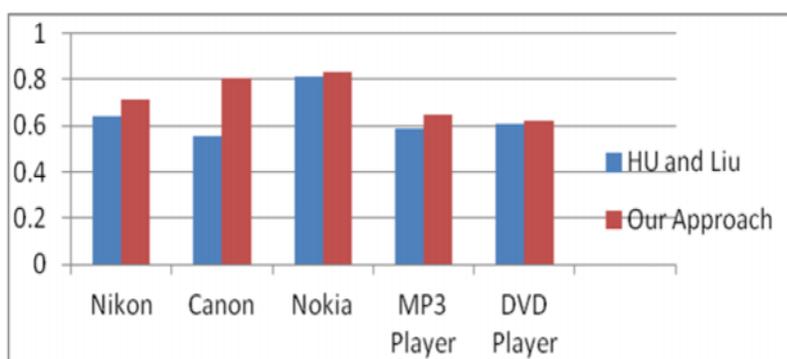


Fig. 7: comparison of Precision V/s Product, between proposed approach to extract opinion sentences with and Minqing Hu et al.[1]

## 5. CONCLUSION AND SCOPE FOR FUTURE

The results of the proposed semantic and dependency driven based approach to extract product features and associated opinion sentences to generate opinion summary about each extracted product feature using different combinations of typed dependencies are effective and promising. It can be improved further by using more combinations of typed dependencies.

## REFERENCES

- [1] Minqing Hu and Bing Liu. "Mining and summarizing customer reviews". In: Proc. of the 10th ACM SIGKDD-2004 international conference on knowledge discovery and data mining, Seattle, pp 168–177
- [2] Kobayashi N, Iida R, Inui K, Matsumotto Y. "Opinion extraction using a learning-based anaphora resolution technique". In Proc. of the second international joint conference on natural language processing (IJCNLP-04), Jeju Island, pp 173–178
- [3] Popescu A, Etzioni O. "Extracting product features and opinions from reviews". In: Proc. of the conference on human language technology and empirical methods in natural language processing, Vancouver, pp 339–346
- [4] Wong TL, Lam W. "Learning to extract and summarize hot item features from multiple auction Web sites". Knowl Inf Syst 14(2)-2008, pp143–160.
- [5] Stanford typed dependencies manual.

- [6] Riloff, E., Janyce, W., Theresa, W. "Learning Subjective Nouns Using Extraction Pattern Bootstrapping" In Proc. 7th Conf. Natural Language Learning, 2003,pp 25-32.
- [7] J.Wiebe. "Learning Subjective Adjectives from Corpora". In Proc. of 12<sup>th</sup> Conference on Innovative Applications of Artificial Intelligence,2000.
- [8] Gamon, M., Aue, A., Corston-Oliver, S., Ringger, E.: "Pulse: Mining Customer Opinions from Free Text". In Proc. 6th Int. Symp. Advances in intelligent data analysis, 005,pp121–132.
- [9] Carenini, G., Ng, R., Pauls, A. " Multi-document Summarization of Evaluative Text". In Proc. 11th Conf. European Chapter of the ACL, 2006.
- [10] Gamgarn Somprasertsri and Pattarachai Lalitrojwong "Mining Feature-Opinion in Online Customer Reviews for Opinion Summarization". Journal of Universal Computer Science, vol. 16, no. 6 (2010),pp 938-955.
- [11] YuanbinWu, Qi Zhang, Xuanjing Huang, LideWu. "Phrase Dependency Parsing for Opinion Mining". Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. Singapore, 6-7 August 2009, pp1533–1541.
- [12] Parma Nand. "On the use of Saliency Weights in Anaphora Resolution". In Proc. of NZCSRSC 2008, April 2008, Christchurch, New Zealand.
- [13] Chih-Ping Wei, Yen-Ming Chen, Chin-Sheng Yang, Christopher and C. Yang. "Understanding what concerns consumers: a semantic approach to product feature extraction from consumer reviews". Springer-Verlag 2009 online
- [14] Theresa Wilson, Janyce Wiebe and Rebecca Hwa. "Just How Mad Are You? Finding Strong and Weak Opinion Clauses". In Proc. of AAI2004.
- [15] Ashequl Qadir. "Detecting Opinion Sentences Specific to Product Features in Customer Reviews using Typed Dependency Relations". Events in Emerging Text Types (eETTs) - Borovets, Bulgaria, pp 38–43.
- [16] V. Hatzivassiloglou and J. Wiebe. " Effects of Adjective Orientation and Gradability on Sentence Subjectivity" In Proc. of the 18<sup>th</sup> conference on Computational linguistics, Germany, 2000
- [17] <http://tartarus.org/~martin/PorterStemmer>
- [18] Agrawal, R. & Srikant, R. 1994. "Fast algorithm for mining association rules". *VLDB'94*, 1994.