*Intelligent Control and Automation*, **2013, 4, 1-10**
http://dx.doi.org/10.4236/ica.2013.41001 Published Online February 2013 (http://www.scirp.org/journal/ica)

# Generating Recommendation Status of Electronic Products from Online Reviews

**Bolanle Adefowoke Ojokoh[*], Olufunke Catherine Olayemi, Olumide Sunday Adewale**
Department of Computer Science, Federal University of Technology, Akure, Nigeria
Email: [*]bolanleojokoh@yahoo.com

Received July 17, 2012; revised October 22, 2012; accepted October 30, 2012

## ABSTRACT

The need for effective and efficient mining of online reviews cannot be overemphasized. This position is as a result of the overwhelmingly large number of reviews available online which makes it cumbersome for customers to read through all of them. Hence, the need for online web review mining system which will help customers as well as manufacturers read through a large number of reviews and provide a quick description and summary of the performance of the product. This will assist the customer make better and quick decision, and also help manufacturers improve their products and services. This paper describes a research work that focuses on mining the opinions expressed on some electronic products, providing ranks or ratings for the features, with the aim of summarizing them and making recommendations to potential customers for better online shopping. A technique is also proposed for scoring segments with infrequent features. The evaluation results using laptops demonstrate the effectiveness of these techniques.

**Keywords:** Opinion Aggregation; Electronic Commerce; Infrequent Feature; Electronic Product; Recommendation

## 1. Introduction

The rapid growth of the web has led to rapid expansion of e-commerce among other things. More customers are turning towards online shopping because it is convenient, reliable, and fast. In order to enhance customer shopping experience, it has become a common practice for online merchants to enable their customers to write reviews on products that they have purchased. Customer reviews of a product are generally considered more honest, unbiased and comprehensive than descriptions provided by the seller. In fact, review comments are one of the most powerful and expressive sources of user preferences. Furthermore, reviews written by other customers describe the usage experience and perspective of (non-expert) customers with similar needs. They give customers a voice, increase consumer confidence, enhance product visibility, and can dramatically increase sales [1,2].

With more and more users becoming comfortable with the Web, an increasing number of people are writing reviews. As a result, the number of reviews that a product receives grows rapidly [3]. Moreover, the consumer reviews are in free form text and consumers prefer to use natural language to express their opinion. It is difficult for a program to "understand" the text information and use these data. Many reviews are long, and as such, it is not an easy task for a potential customer to make a decision whether or not to purchase a product based on the reviews he reads. It is therefore a very important and challenging problem to mine these reviews and produce a summary of them and also propose a recommendation decision to the potential customer.

A host of research works have been proposed to profer solution to problems related to mining and summarizing customer reviews [4,5], which is called *opinion mining* or *sentiment analysis*. Opinion Mining has two main research directions, document level opinion mining and feature level opinion mining [6]. Document level mining involves determining the document's polarity by calculating the average semantic orientation of extracted phrases. In feature level opinion mining, reviews are summarized and classified by extracting high frequency feature and opinion keywords. Feature-opinion pairs are identified by using a statistical approach or labeling approach and dependency grammar rules to handle different kinds of sentence structures. Generally, feature level opinion mining has greater precision over the document level and uses basically product features in analysis and evaluation. The main tasks present in most past and current research works are: to find product features that have been commented on by reviewers [7] and to decide whether the comments are positive, negative or neutral [3]. Nevertheless, it is important to discover how positive or negative the comments are and further, make a concise decision about the product (recommended or not) to the poten-

[*]Corresponding author.

tial customer. This work makes a contribution by classifying the opinions about each feature, hence showing how negative or positive it is. It also proposes a technique to include comments on infrequent features into the recommendation decision for electronic products.

The remainder of this paper is organized as follows. Section 2 contains a summary of related works. We describe the proposed technique in Section 3. Section 4 contains the experiments and evaluation, while Section 5 concludes the work and presents proposed future research work.

## 2. Related Work

Opinion mining has been studied by many researchers in past years. The earliest research on opinion mining was on identifying opinion (or sentiment) bearing words. Hatzivassiloglou and McKeown [8] identified several linguistic rules that can be exploited to identify opinion words and their orientations from a large corpus. The work was applied, extended and improved in [9].

Another major development in the area of opinion mining is sentiment classification of product reviews at the document level [10]. The objective of this task is to classify each review document as expressing a positive or a negative sentiment about an object (such as a movie, camera, car).

Feature-based opinion mining and summarization have been proposed by a number of researchers. Hu & Liu [7] proposed a technique based on association rule mining to extract product features. The main idea is that people often use the same words when they comment on the same product features. Then frequent itemsets of nouns in reviews are likely to be product features while the infrequent ones are less likely to be product features. This work also introduced the idea of using opinion words to find additional (often infrequent) features. Liu *et al.* [9] improved upon Hu's work by proposing a technique based on language pattern mining to identify product features from pros and cons in reviews in the form of short sentences. They also made an effort to extract implicit features. Moreover, [11] proposed feature extraction for capturing knowledge from product reviews. The output of Hu and Liu's system served as input to their system, and the input was mapped to the user-defined taxonomy features hierarchy thereby eliminating redundancy and providing conceptual organization. To identify the expressions of opinions associated with features, Hu & Liu focused on adjacent adjectives that modify feature nouns or noun phrases. They used adjacent adjectives as opinion words that are associated with features.

Popescu and Etzioni [12] investigated the same problem. Their algorithm requires that the product class is known. The algorithm determines whether a noun/noun phrase is a feature by computing the pointwise mutual information (PMI) score between the phrase and class specific discrimination. This work first used part-whole patterns for feature mining, but it finds part-whole based features by searching the Web and querying the Web is time-consuming. Qiu *et al.* [13] proposed a double propagation method, which exploits certain syntactic relations of opinion words and features, and propagates through both opinion words and features iteratively. The extraction rules are designed based on different relations between opinion words and features, and among opinion words and features themselves. Dependency grammar was adopted to describe these relations.

Another related work, but using a bit different approach is that of [5]. They proposed a method to extract product features from user reviews and generate a review summary, using product specifications rather than resources like segmenter, POS tagger or parser. At the feature extraction stage, multiple specifications are clustered to extend the vocabulary of product features. Hierarchy structure information and unit of measurement information are mined from the specification to improve the accuracy of feature extraction. At summary generation stage, hierarchy information in specifications is used to provide a natural conceptual view of product features.

Another area is mining of comparative sentences. It basically consists of identifying what features and objects are compared and which objects are preferred by their authors (opinion holders) [14]. Another related work involving comparison is that of [15] that compare reviews of different products in one category to find the reputition of the target product. However, it does not summarize reviews, and it does not mine product features on which the reviewers have expressed their opinions. Although they do find some frequent phrases indicating reputations, these phrases may not be product features (for example, "doesn't work", "benchmark result" and "no problem(s)").

Feng *et al.* [16] and Ojokoh and Kayode [17] also proposed relateds method to solve the problem of opinion mining and applied this for some electronic products from amazon. There are a few differences between the works and ours. The method for determing the opinion polarity and aggregation is different from ours. Moreover, [16] only stopped at classifying the aggregated opinion as negative or positive and adopted that for recommendation. Also, their work did not consider infrequent feature identification. This works proposes the adoption of a threshold value for recommendation based on experiments with amazon rated views.

## 3. The Proposed Method

**Figure 1** gives the architectural overview of the proposed system. In summary, the system executes the following steps:

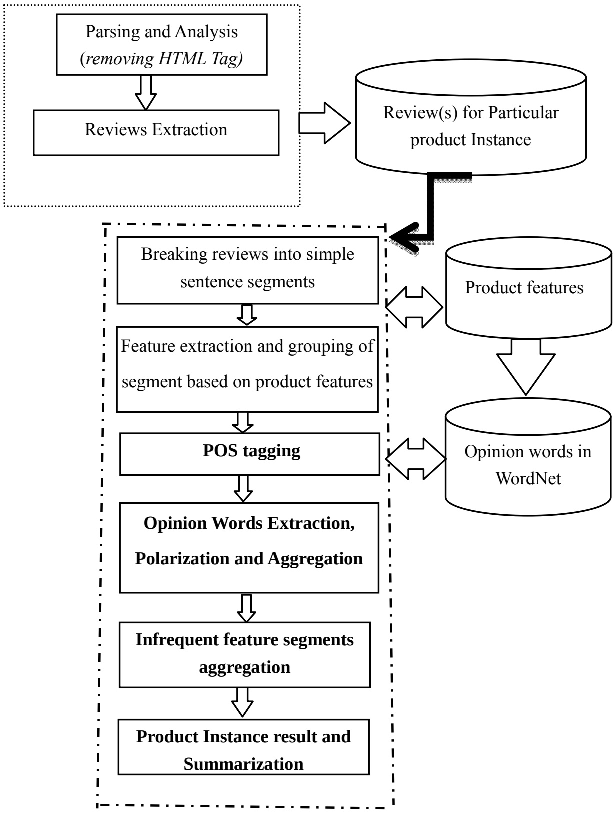

Parsing and Analysis
(*removing HTML Tag)*
Reviews Extraction
Review(s) for Particular product Instance
Breaking reviews into simple sentence segments
Product features
Feature extraction and grouping of segment based on product features
POS tagging
Opinion words in WordNet
Opinion Words Extraction, Polarization and Aggregation
Infrequent feature segments aggregation
Product Instance result and Summarization

**Figure 1. Architecture of the proposed system.**

1) Download online product reviews;

2) Categorize review(s) into segments based on product features;

3) Classify review segments based on polarity;

4) Score the obtained polarized review segments;

5) Incorporate infrequent feature segments;

6) Make recommendations to potential customers in an online shop.

Given a product class, *C* with instances *I* and reviews *R*, the goal of the system is to find the set of (feature, opinions) tuples $\{f, o_1, \cdots, f, o_j\}$ s.t $f\ \varepsilon\ F$ and $o_1,\cdots, o_j\ \varepsilon$ *O*, where:

*F* is a set of product class features in *R*;

*O* is a set of opinion phrases in *R*;

*f* is a feature of a particular product instance;

*o* is an opinion about *f* in a particular sentence.

These steps as reflected in the architectural overview are described in details as follows.

## 3.1. Preprocessing

The HTML files containing online reviews downloaded from a product review website [18] were analyzed and parsed to extract useful information and review(s) about every product instance present. Java HTML Parser was used for analysis and extraction of the reviews. Java HTML Parser is a java library used to parse HTML files in either a linear or nested fashion, and it is primarily used for extraction. In this process, the HTML tags in the review documents are removed.

The needed review(s) are then extracted and stored in the database. A list of product features for a particular product class is maintained in the database to enable categorization of the extracted reviews into feature segments.

Each review is then disintegrated into simple sentence segments using punctuations such as ", : ; .". The segmented reviews are analysed and categorized based on the feature set found in the feature database. Each segment category for every product instance is passed to the POS Tagger.

## 3.2. Part of Speech (POS) Tagging

Every word in each segment category is parsed and tagged by part of speech using Stanford parser. Stanford parser [19] is a piece of software that reads text in some languages and assigns parts of speech to each word (and other tokens), such as noun (NN), verb (VBZ), adjective (JJ), noun phrase (NP), verb phrase (VP), adjective phrase (ADJP), adverb (RB) etc and the root sentence (S).

To parse a sentence like "battery life is extremely poor", the outcome is given in **Figure 2**.

The parts of speech we are interested in are nouns (NN) and adjectives (JJ), since they correspond to feature and opinion pairs that are to be extracted. However, some words could not be extracted independently; for example, considering tagging the sentence like:

"*This product is not too bad*", where customer's opinion is negated by the adverb "not". Therefore, the idea proposed in [20] was employed. Two consecutive words are extracted from the review segments if their tags conform to any of the patterns in **Table 1**.

In **Table 1**, JJ tags indicate adjectives, the NN tags are nouns, the RB tags are adverbs, and the VB tags are verbs. The first pattern from the table, means that two consecutive words are extracted if the first word is an adjective and the second is a noun; the second pattern means that two consecutive words are extracted if the first word is an adverb and the second word is adjective, but the third word—which is not extracted—cannot be a noun; the third pattern means that two consecutive words are extracted if they are both adjectives but the following word is not noun. Singular and plural proper nouns are avoided, so that the names of items in the review cannot

```
(ROOT
  (S
    (NP (NN battery) (NN life))
    (VP (VBZ is)
      (ADJP (RB extremely) (JJ poor)))))
```

**Figure 2. Hierarchical tree output from Stanford parser.**

**Table 1. Feature-opinion extraction pattern.**

| Word 1 | Word 2 | Word 3 |
|---|---|---|
| JJ | NN/NNS | Anything |
| RB/RBR/RBS | JJ | Not NN or NNS |
| JJ | JJ | Not NN or NNS |
| NN/NNS | JJ | Not NN or NNS |
| RB/RBR/RBS | VB/VBN/VBD/VBG | Anything |

influence the classification.

### 3.3. Opinion Words Extraction, Polarization and Aggregation

Opinion words are the words people use to express a positive or negative comment about a subject. In this work, a lexicon—based approach is used (based on previous work on the correlation of subjectivity and the presence of adjectives in sentences [21,22] where opinion words are assumed to be adjectives). The list, containing 2006 positively and 4783 negatively oriented adjectives, created by [7], is used for this purpose.

An aggregated score [3] is obtained for each segment. The summation of the scores obtained for segments in this group forms the grade for the feature. The features grade is used in turn to summarize the instance.

The opinions obtained are identified and classified based on the review phrase whose key word has a negative, positive or neutral semantic orientation. A value, feature-opinion word score, is assigned to the feature for the segment where feature and opinion word is found. The summation of this value determines the score for the feature of an instance.

### 3.4. Infrequent Feature Segments Aggregation

Segments that are referred to as "Infrequent feature segments" by our system are the ones with comments that are associated with silent features, or non-identifiable ones such as "*This is an excellent laptop*". In this case, there are no specific features that such opinion words are associated with. We discover in this work that they still need some consideration rather than being considered as too significant. Hence, before a final recommendation decision is made for any product, our system proposes the inclusion of some values for the infrequent feature segments to form the final summation value that determines if a final recommendation will be made or not. Therefore, the aggregation function proposed by [3] is modified to accomodate infrequent feature segments as follows.

Given lists of positive, negative and context dependent opinion words, including phrases and idioms, the opinion aggregation function for frequent features is calculated using Equation (1).

$$\text{score}\left(f_i, s_k\right) = \sum_{w_j \in s_k, v} \frac{w_j \cdot SO}{dis\left(w_j, f_i\right)}; \quad i = 1,\cdots,m; j = 1,\cdots,n; k = 1,\cdots,t \tag{1}$$

where $w_j$ is an opinion word, $V$ is the set of all opinions in our list, $s_k$ is the sentence segment that contains the feature $f_i$, and $dis(w_j, f_i)$ is the distance between feature $f_i$ and opinion word $w_j$ in $s_k{\cdot}w_j{\cdot}SO$ is the semantic orientation of $w_i$. If the final score is positive, then the opinion on the feature in the sentence $s$ is positive. If the final score is negative, then the opinion on the feature is negative, otherwise it is considered to be neutral. The multiplicative inverse in the formula is used to give low weights to opinion words that are far away from feature $f_i$. A positive word is assigned the semantic orientation score of 1, and a negative word is assigned the semantic orientation score of −1. In Hu & Liu (2004), a simple summation is used. The new function is better because far away opinion words may not modify the current feature.

Further, a distinction is made between the positive and negative segments scores as shown in Equations (2) and (3):

$$P\left(f_i\right) = \sum_{j=1}^{n} P_j\left(\text{score}\left(f_i, s_k\right)\right) \tag{2}$$

$P_j\left(\text{score}\left(f_i, s_k\right)\right)$ represents positive feature score for a given segment $s_k$ that contains the feature $f_i$. The sum of all the positive scores obtained from segment $s_k$ for a particular feature is represented by $P\left(f_i\right)$ and $n$ is the total number of positive score segments.

$$N\left(f_i\right) = \sum_{j=1}^{n} N_j\left(\text{score}\left(f_i, s_k\right)\right) \tag{3}$$

$N_j\left(\text{score}\left(f_i, s_k\right)\right)$ represents negative feature score for a given segment $s_k$ that contains the feature $f_i$. The sum of all the negative scores obtained from segment $S_k$ for a particular feature is represented by $N\left(f_i\right)$, where $n$ is the total number of negative score segments.

An aggregated score is obtained for every feature, $f_i$, as shown in Equation (4):

$$Agg\left(f_i\right) = P\left(f_i\right) + \left|N\left(f_i\right)\right| \tag{4}$$

This work also proposes a percentage value to be used to determine the quality of product feature $f_i$ from reviews $S$. The higher the positive percentage value the better the product feature, and the higher the negative percentage value, the lower the quality. These are described in Equations (5) and (6).

$$PV\left(f_i\right) = \frac{P\left(f_i\right)}{Agg\left(f_i\right)} \times 100 \tag{5}$$

$PV(f_i)$ is the aggregated positive feature score in percentage

$$NV(f_i) = 100 - PV(f_i) \qquad (6)$$

$NV(f_i)$ is the aggregated negative feature score in percentage.

This work proposes a method for the inclusion of infrequent feature segments, with some explanations. It was noticed that with many of the datasets, the infrequent feature segments have some effect on the results, especially when they are negative segments. With some experiments, it was also discovered, that as the number increases (if is positive) or decreases (if it is negative), the effect reduces. So, the technique proposed was to reduce the effect of the inclusion of these segments with increase in the number for negative segments and a threshold was set for positive segments.

The proposed technique is proposed in the algorithm in **Table 2**.

**Table 2. Opinion aggregation for infrequent feature segments.**

```
Algorithm Opinion Infrequent Aggregation ()

$Inf(S_k) = nP(\inf) - nN(\inf)$ /* $nP(\inf)$ and $nN(\inf)$ are the number of positive and negative segments */
if $Inf(S_k) < 0$ /* $Inf(S_k)$ is the infrequent segment score */
    if $\left(Inf(S_k) \le \frac{PV(f_i)}{3}\right)$
        $PV(f_i) = PV(f_i) + Inf(S_k)$
    else
        $PV(f_i) = PV(f_i) + Inf(S_k)/2$
    $NV(f_i) = 1 - PV(f_i)$
    endif
else
    if $Inf(S_k) > 0$
        if $\left(Inf(S_k) \le \frac{NV(f_i)}{3}\right)$
            $PV(f_i) = PV(f_i) + Inf(S_k)$
        else if $Inf(S_k) > 100$ /* $PV(f_i) +$ is the re-aggregated positive feature score in percentage */
            $PV(f_i) = PV(f_i) + Inf(S_k)/2$
        else $PV(f_i) = PV(f_i) + 100/3$
            $NV(f_i) = 100 - PV(f_i)$
        endif
    endif
if (no_of_frequent_feature $\le 1$) /* infrequent features is more important*/
$P(f_i) = \sum_{j=1}^{n} P_j\left(\text{score}(f_i, s_k) + 0.6\sum_{k=1}^{m} Inf(S_k)\right)$
endif /*where m is the number of infrequent feature segments */
```

The algorithm proposed in [3] was modified to incorporate our idea of review classification and recommendation decision inclusion. It is described in the algorithm in **Table 3**.

This work also proposes a ranking system for the feature scores. This is reasonable, because many comments on some of the popular electronic commerce are rated. For instance, amazon [23] awards stars (1 to 5; the higher, the better) to reviews of different products. From this ranking, the features scores and remarks are obtained. The method used is shown in **Table 4**. So from the aggregated value obtained previously, for feature, $f_i$ a

**Table 3. Predicting the orientations of opinion on product features.**

```
Algorithm Opinion Orientation ()

foreach sentence Si that contains a set of features do
        positive_orientation=0
        negative_orientation=0
    foreach segment ($S_k$ )in $S_i$ that contain feature $f_i$ do
        feature ($f_i$) = features contained in $S_k$
        orientation = wordOrientation($ow$, $f_i$, $S_k$)
        orientation > 0
                $f_i$'s orientation in $sm$ = 1
                positive_orientation += orientation
        else if orientation < 0
                $f_i$'s orientation in sm = −1
        negative_orientation+= orientation
        else
                $f_i$'s orientation in $sm$ = 0
    endif
    endfor
            aggregate score for feature $f_i$ = positive_orientation + $|negativeorientation|$
endfor

    Procedure word Orientation (word, feature, segment)
    if word is a Negation word
            orientation = −1;
        mark words in segment
  elseif word is a NOT word
            orientation = change word orientation;
            mark words in segment
  else
        orientation = 1
        mark words in segment
endif
orientation= $\frac{\text{orientation}}{dis(\text{word,feature})}$
return orientation
```

**Table 4. Feature score value and remark.**

| Range | Rank or Rating Value | Remark |
|---|---|---|
| $PV(f_i) \ge 80$ | 5 | Excellent |
| $80 > PV(f_i) \ge 60$ | 4 | Very Good |
| $60 > PV(f_i) \ge 50$ | 3 | Good |
| $50 > PV(f_i) \ge 40$ | 2 | Fair |
| $PV(f_i) < 40$ | 1 | Poor |

feature score value and corresponding remark are generated.

### 3.5. Product Instance Result and Summarization

The recommendation status is derived, based on the threshold (60%) fixed for recommendation. If the positive sentiment score is greater than or equal to the threshold, it means the authors recommend the product. Otherwise, it means the authors do not recommend this product. For instance, if the recommendation summary generated is:

Positive: 90%

Negative: 10%

The Recommendation Status is "***Recommended***".

The threshold of 60% was arrived at after performing experiments with some rated downloaded reviews from amazon, in addition to the classification decision adopted by [16]. It is reasonable to assume that the 4 star and 5 star reviews should lead to a positive recommendation, while 1, 2 or 3 star should lead to negative recommendation (not recommended).

## 4. Experiments and Evaluation

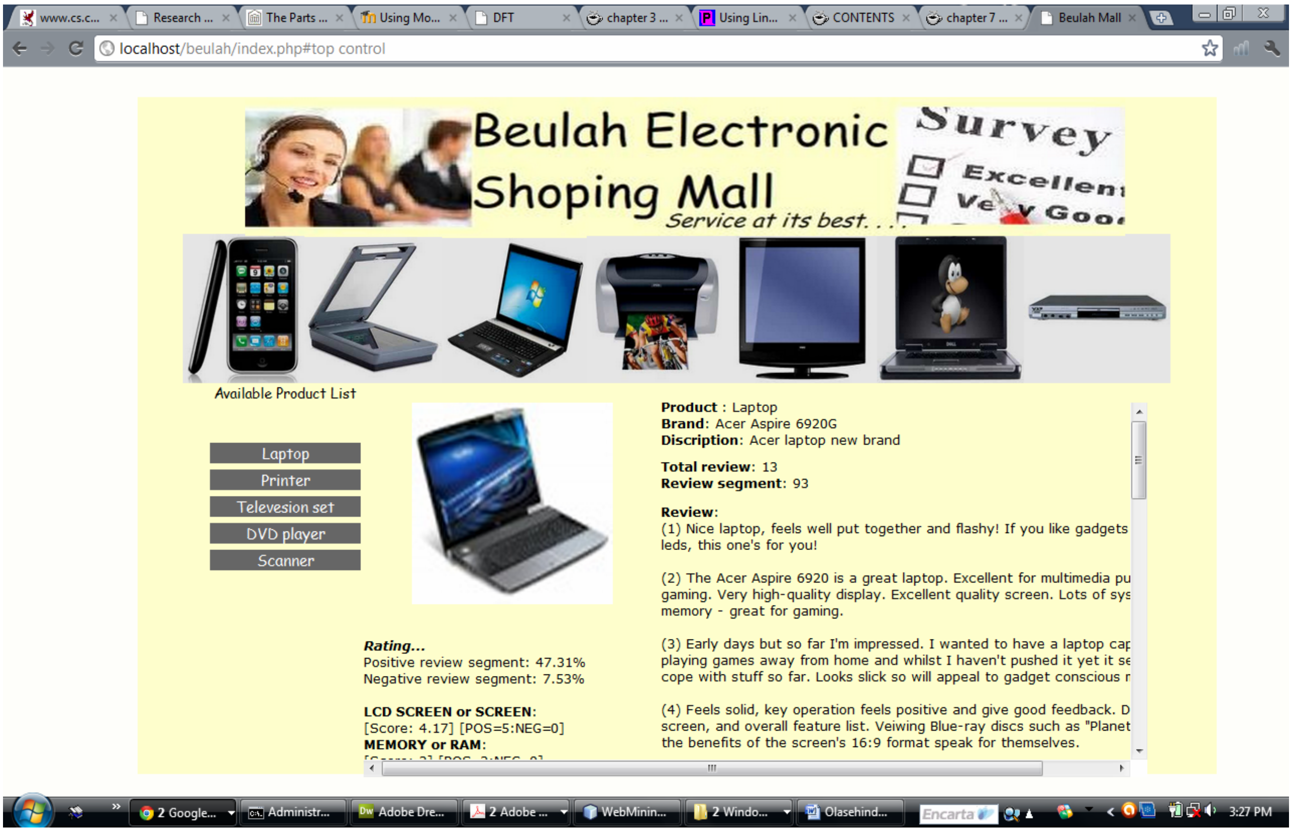
**Figures 3** and **4** show the screen shots of the developed online shopping mall, where our opinion mining application was implemented. **Figure 3** is the home page, while **Figure 4** is the web page that provides the means to display the reviews of the different electronic products used in the system. The product classes used for the experiments are laptop, printer, scanner, DVD player and television set, with 59 product instances in all. Several user reviews downloaded from an online review website [18] were used for the experiments. The characteristics of these data are shown in **Table 5**.

Experiments were carried out with online reviews of different brands (instances) of DVD players, laptops, scanners, printers and television sets. Some sample experimental results are shown in **Tables 6-9**.

Experiments were carried out to finally arrive at the threshold of 60%. It was discovered with the trend of the reviews downloaded from amazon and from Feng *et al.* (2010)'s adoption of 4 and 5 star reviews of amazon as positive reviews and the others (1 to 3) as negative, that it is reasonable to set the threshold for recommendation as 60% (as our evaluation results affirm).

### Evaluation

In order to evaluate the performance of the proposed system, 499 laptops rated reviews were downloaded from amazon to test the accuracy of the classification of the reviews and the final recommendation decision made by the system. **Table 10** gives an analysis of the number of



**Product** : Laptop
**Brand**: Acer Aspire 6920G
**Discription**: Acer laptop new brand

**Total review**: 13
**Review segment**: 93

**Review**:
(1) Nice laptop, feels well put together and flashy! If you like gadgets leds, this one's for you!

(2) The Acer Aspire 6920 is a great laptop. Excellent for multimedia pu gaming. Very high-quality display. Excellent quality screen. Lots of sys memory - great for gaming.

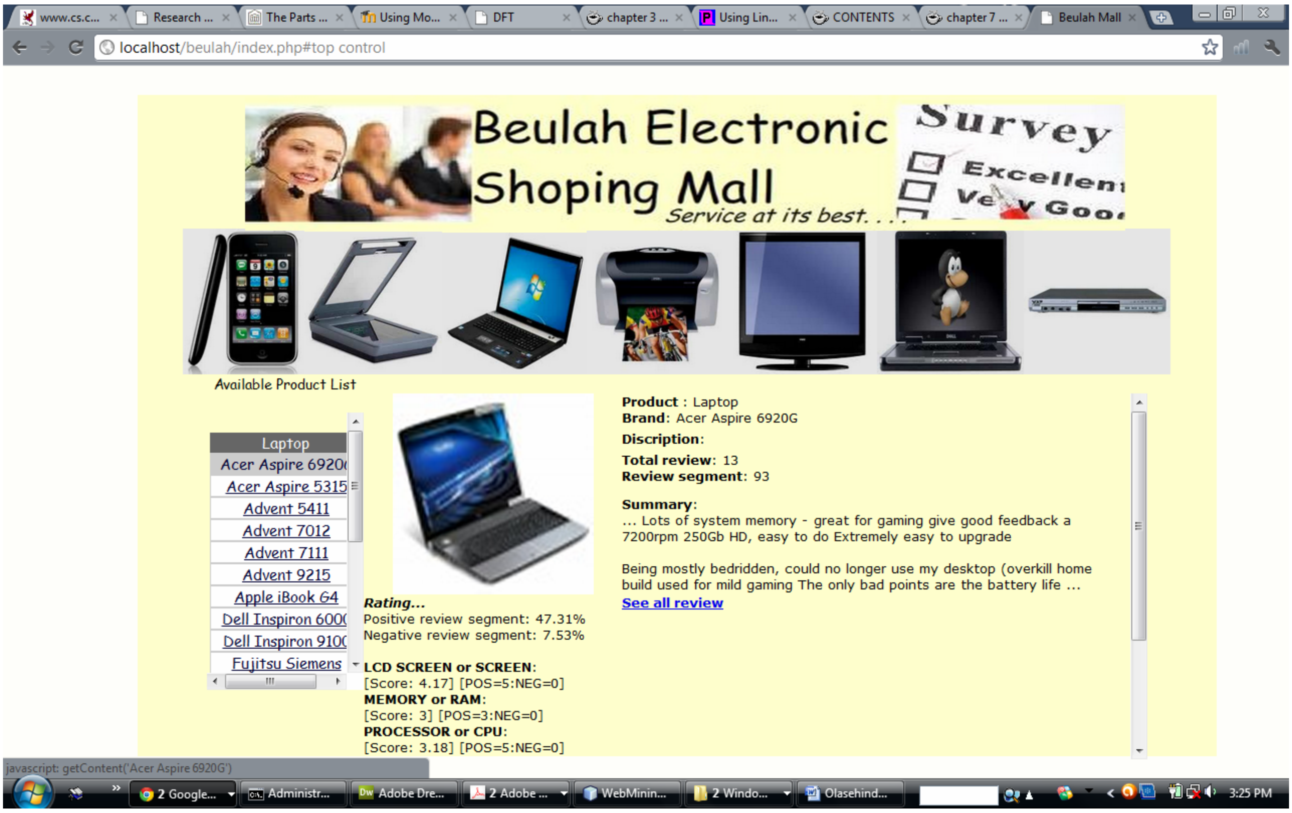
(3) Early days but so far I'm impressed. I wanted to have a laptop cap playing games away from home and whilst I haven't pushed it yet it se cope with stuff so far. Looks slick so will appeal to gadget conscious r

(4) Feels solid, key operation feels positive and give good feedback. D screen, and overall feature list. Veiwing Blue-ray discs such as "Planet the benefits of the screen's 16:9 format speak for themselves.

***Rating...***
Positive review segment: 47.31%
Negative review segment: 7.53%

**LCD SCREEN or SCREEN**:
[Score: 4.17] [POS=5:NEG=0]
**MEMORY or RAM**:

**Figure 3. Online shopping mall home page.**



**Product** : Laptop
**Brand**: Acer Aspire 6920G

**Discription**:

**Total review**: 13
**Review segment**: 93

**Summary**:
… Lots of system memory - great for gaming give good feedback a 7200rpm 250Gb HD, easy to do Extremely easy to upgrade

Being mostly bedridden, could no longer use my desktop (overkill home build used for mild gaming The only bad points are the battery life …

See all review

***Rating...***
Positive review segment: 47.31%
Negative review segment: 7.53%

**LCD SCREEN or SCREEN**:
[Score: 4.17] [POS=5:NEG=0]
**MEMORY or RAM**:
[Score: 3] [POS=3:NEG=0]
**PROCESSOR or CPU**:
[Score: 3.18] [POS=5:NEG=0]

**Figure 4. Online shopping mall review page.**

**Table 5. Characteristics of the review data.**

| Product Name | No. of Reviews | No. of Features |
|---|---|---|
| Laptop | 71 | 21 |
| Printer | 39 | 21 |
| DVD Player | 36 | 16 |
| Scanner | 14 | 10 |
| Television Set | 235 | 18 |

the reviews used for the evaluation experiments in terms of the number of ratings by amazon. We adopted the method described above in order to rate our reviews into five ranks like amazon, and tried to evaluate how the system performs. The set of reviews that are eventually given the same rank as amazon does is termed "correctly ranked" and "not correctly ranked" otherwise. **Table 11** shows the results of this evaluation. It can be seen from the results that incorporating infrequent feature segments results contributes significantly to the improvement of the accuracy of the system for review classification.

**Table 12** shows the result of the final recommendation result, which is the actual goal of the proposed system. The evaluation conducted here follows the method adopted by [16]. For all the product instances, any 4 or 5 star review that receives a recommendation decision by our system is termed correct, while any 1, 2 or 3 star review that receives a recommendation decision is termed incorrect. The proposed system performs significantly better than some related systems in making recommendation decision. For instance, our proposed system makes 100% accuracy for three products out of five and 60% and 80% for the others without incorporating infrequent feature segments results. The recommendation process achieves 100% accuracy with the tested data sets for all the electronic products after implementing our proposed technique for incorporating infrequent feature segments. The accuracy of the ranking done by our system is shown in **Table 13**. It achieves 100% accuracy for ranks 1, 3 and 5. The accuracy for ranks 2 and 4 are relatively low. However, for the missing places in rank four, the system classifies most of the results under rank 5, which will lead to a correct recommendation decision at the end of the day. The low results for rank two are mostly for the Acer and Toshiba Sattelite datasets. The reasons are not clearly known. Many existing systems used their methods on clearly different datasets and different products, from the ones used in the system. However, according to the published results of one of the nearest systems [16] (shown in **Table 14**), for the same decision (but for different products), 85%, 84%, 70%, 78% accuracy were obtained for the four different products they evaluated on. Although the products are different, this still shows that our system is effective, especially at making a final recommendation decision.

**Table 6. Product: DVD player instance name: Bush DVD2023.**

| Feature name | No. of positive reviews | No. of negative reviews | Feature score | Rank | Remark |
|---|---|---|---|---|---|
| Magnetic fields and heat | 0 | 2 | −1.5 | 1 | Poor |
| Video output | 0 | 1 | −1 | 1 | Poor |
| Audio output | 1 | 0 | 1 | 2 | Fair |
| Digital quality | 1 | 0 | 1 | 2 | Fair |
| **Infrequent features** | 21 | 8 | 13 | 5 | Excellent |
| **Total/Recommendation** | **2** | **3** | **POS = 45% NEG = 55%** | **2** | **Not Recommended** |

**Table 7. Product: Laptop instance name: Apple iBook G4.**

| Feature name | No. of positive reviews | No. of negative reviews | Feature score | Rank | Remark |
|---|---|---|---|---|---|
| LCD Screen or Screen | 3 | 0 | 2.5 | 3 | Good |
| Hard Drive or HD or HDD | 4 | 3 | 0.5 | 1 | Poor |
| Keyboard | 0 | 1 | −0.08 | 1 | Poor |
| Memory or RAM | 3 | 0 | 3 | 4 | V. Good |
| Mother board or System board | 2 | 3 | −0.86 | 1 | Poor |
| Video card or Graphics card | 4 | 0 | 4 | 4 | V. Good |
| Battery | 2 | 0 | 2 | 3 | Fair |
| **Infrequent features** | 58 | 17 | 41 | 5 | Excellent |
| **Total/Recommendation** | **18** | **7** | **POS = 93% NEG = 7%** | **5** | **Recommended** |

**Table 8. Product: Scanner instance name: HP 3200C.**

| Feature name | No. of positive reviews | No. of negative reviews | Feature score | Rank | Remark |
|---|---|---|---|---|---|
| Resolution | 0 | 1 | −0.11 | 1 | Poor |
| Price | 2 | 0 | 1.33 | 2 | Fair |
| Image | 1 | 2 | −1.67 | 1 | Poor |
| Software | 0 | 1 | −1 | 1 | Poor |
| **Infrequent features** | 6 | 6 | 0 | 1 | Poor |
| **Total/Recommendation** | **3** | **4** | **POS = 32% NEG = 68%** | **1** | **Not Recommended** |

**Table 9. Product: Television set instance name: Sharp Aquos LC42XD1E LCD.**

| Feature name | No. of positive reviews | No. of negative reviews | Feature score | Rank | Remark |
|---|---|---|---|---|---|
| Price or Cost | 3 | 0 | 2.33 | 3 | Good |
| TV settings and modes | 4 | 2 | 0.2 | 1 | Poor |
| LED backlighting | 0 | 1 | −0.33 | 1 | Poor |
| Display or Picture | 5 | 0 | 3.39 | 4 | V. Good |
| **Infrequent feature** | 11 | 3 | 8 | 5 | Excellent |
| **Total/Recommendation** | **12** | **3** | **POS = 95% NEG = 5%** | **5** | **Recommended** |

**Table 10. Analysis of reviews used for evaluation experiments.**

| Rating | No. of reviews |
|---|---|
| One Star | 53 |
| Two Stars | 36 |
| Three Stars | 46 |
| Four Stars | 112 |
| Five Stars | 252 |

## 5. Conclusion and Future Work

In this paper, we propose a set of techniques for mining and classifying opinions expressed on the features of products as recommended or not recommended. It provides a method for scoring review segments containing “infrequent features”. It also proposes the inclusion of a rank or rating for each feature like other electronic com-

**Table 11. The evaluation result of review rating.**

| Product name | No. of reviews | Accuracy without infrequent segment results | Accuracy incorporating infrequent segment results |
|---|---|---|---|
| Apple MacBook MC516LL A 13 | 101 | 0.75 | 0.75 |
| Toshiba NB505-N508BN | 210 | **0.4** | **1.0** |
| Acer AS5253-BZ602 | 47 | **0.4** | **0.6** |
| Toshiba Satellite L755-S5271 | 75 | **0.2** | **0.6** |
| Dell 600 | 66 | **0.4** | **1.0** |

**Table 12. The evaluation result of recommendation decision.**

| Product name | No. of reviews | Accuracy without infrequent segment results | Accuracy incorporating infrequent segment results |
|---|---|---|---|
| Apple MacBook MC516LL A 13 | 101 | 1.0 | 1.0 |
| Toshiba NB505-N508BN | 210 | **0.6** | **1.0** |
| Acer AS5253-BZ602 | 47 | 1.0 | 1.0 |
| Toshiba Satellite L755-S5271 | 75 | 1.0 | 1.0 |
| Dell 600 | 66 | **0.8** | **1.0** |

**Table 13. The accuracy of the ranks/ratings.**

| Rating | No. of reviews | Accuracy |
|---|---|---|
| One Star | 53 | 1.0 |
| Two Stars | 36 | 0.5 |
| Three Stars | 46 | 1.0 |
| Four Stars | 112 | 0.4 |
| Five Stars | 252 | 1.0 |

**Table 14. The evaluation result of review classification [16].**

| Product Name | No. of reviews | Accuracy |
|---|---|---|
| MacBook MC516LL A 13 | 138 | 0.85 |
| Motorola V3 | 147 | **0.84** |
| BlackBerry | 114 | 0.7 |
| Acer Aspire | 254 | 0.78 |

merce sites do. It finally aggregates all these to make a recommendation decision using an experimental threshold value for potential buyers of electronic products in an online shopping mall. The experimental results indicate that the proposed techniques are effective at performing this task. In the future, the work can be extended to test our proposed algorithm on other data sets, and also improve on the methods adopted in this work for automatic tag identification on the review web pages, to handle more complex sentences, especially those written in improperly structured English language.

## REFERENCES

[1] L. Zhang, B. Liu, S. H. Lim, S. H. and E. O'Brien-Strain, "Extracting and Ranking Product Features in Opinion Documents," *Proceedings of the* 23rd *International Conference on Computational Linguistics* (*COLING*), Beijing, 23-27 August 2010, pp. 1462-1470.

[2] S. Aciar, "Mining Context Information from Consumer's Reviews," *Proceedings of the* 2nd *Workshop on Contex-Aware Recommender Systems*, Barcelona, 26 September 2010.

[3] X. Ding, B. Liu and P. S. Yu, "A Holistic Lexicon-Based Approach to Opinion Mining," *Proceedings of WSDM* 08, *California*, Palo Alto, 11-12 February 2008, pp. 231-239. doi:10.1145/1341531.1341561

[4] G. Somprasertsri and P. Lalitrojwong, "Mining Feature-Opinion in Online Customer Reviews for Opinion Summarization," *Journal of Universal Computer Science*, Vol. 16, No. 6, 2009, pp. 938-955.

[5] X. Meng and H. Wang, "Mining User Reviews: From Specification to Summarization," *Proceedings of the ACL-IJCNLP* 2009 *Conference Short Papers*, 2009, pp. 177-180. doi:10.3115/1667583.1667637

[6] W. Jin and H. H. Ho, "A Novel Lexicalized HMM-Based Learning Framework for Web Opinion Mining," *Proceedings of the* 26th *International Conference on Machine Learning*, Montreal, 14-18 June 2009.

[7] M. Hu and B. Liu, "Mining and Summarizing Customer Reviews," *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, 22-25 August 2004, pp. 168-177.

[8] V. Hatzivassiloglou and K. McKeown, "Predicting the Semantic Orientation of Adjectives," *Proceedings of the* 35*th Annual Meeting of the Association for Computational Lingusitics*, Madrid, 7-10 July 1997, pp. 174-181. doi:10.3115/976909.979640

[9] B. Liu, M. Hu and J. Cheng, "Opinion Observer: Analyzing and Comparing Opinions on the Web. *Proceedings of International World Wide Web Conference* (*WWW*'05), New York, 2005, pp. 342-351. doi:10.1145/1060745.1060797

[10] J. Wiebe and E. Riloff, "Creating Subjective and Objec-

tive sentence classifiers from unannotated texts," *Proceedings of International Conference on Intelligent Text Processing and Computational Linguistics* (*CICLing*'05), Mexico City, 13-19 February 2005, pp. 486-497. doi:10.1007/978-3-540-30586-6_53

[11] G. Carenini, R. T. Ng and E. Zwart, "Extracting Knowledge from Evaluative Text," *Proceedings of the Third International Conference on Knowledge Capture*, Banff, 2-5 October 2005, pp. 11-18. doi:10.1145/1088622.1088626

[12] M. Popescu and O. Etzioni, "Extracting Product Features and Opinions from Reviews," *Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP* '05, Vancouver, 6-8 October 2005, pp. 339-346. doi:10.3115/1220575.1220618

[13] G. Qiu, B. Liu, J. Bu and C. Chen, "Expanding Domain Sentiment Lexicon through Double Propagation," *International Joint Conferences on Artificial Intelligence*, Pasadena, 11-17 July 2009.

[14] G. Ganapathibhotla and B. Liu, "Identifying Preferred Entities in Comparative Sentences," *Proceedings of the 22nd International Conference on Computational Linguistics* (*COLING*'08), Manchester, 2008.

[15] S. Morinaga, K. Yamanishi, K. Tateishi and T. Fukushima, "Mining Product Reputations on the Web," *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, 23-26 July 2002.

[16] S. Feng, M. Zhang and Y. Zhang, "Recommended or Not Recommended? Review Classification through Opinion Extraction," *Proceedings of APWEB Conference*, Busan, 6-8 April 2010.

[17] B. A. Ojokoh and O. Kayode, "A Feature-Opinion Extraction Approach to Opinion Mining," *Journal of Web Engineering*, Vol. 11, No. 1, 2012, pp. 051-063.

[18] http://reviewcenter.com

[19] Stanford Parser. http://nlp.stanford.edu/software/lex-parser.shtml

[20] P. D. Turney, "Thumbs up or Thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews," *Proceedings of the Conference of the Association of Computational Lingustics*, Howard, 24 August-1 September 2002, pp. 417-424.

[21] J. Wiebe, R. Bruce and T. O'Hara, "Development and Use of a Gold Standard Data Set for Subjectivity Classifications," *Proceedings of the Conference of the Association of Computational Lingustics*, College Park, 20-26 June 1999.

[22] R. Bruce and J. Wiebe, "Recognizing Subjectivity: A Case Study of Manual Tagging," *Natural Language Engineering*, Vol. 5, No. 2, 2000 pp. 187-205. doi:10.1017/S1351324999002181

[23] Amazon Web Service. http://www.aws.amazon.com