

# Mining Feature-Opinion Pairs and Their Reliability Scores from Web Opinion Sources

Ahmad Kamal  
Department of Mathematics  
Jamia Millia Islamia  
New Delhi, India  
ahmad.kamal77@gmail.com

Muhammad Abulaish\*  
CoE in Information Assurance  
King Saud University  
Riyadh, Saudi Arabia  
mAbulaish@ksu.edu.sa

Tarique Anwar  
CoE in Information Assurance  
King Saud University  
Riyadh, Saudi Arabia  
tAnwar.c@ksu.edu.sa

## ABSTRACT

Due to proliferation of Web 2.0, there is an exponential growth in user generated contents in the form of customer reviews on the Web, containing precious information useful for both customers and manufacturers. However, most of the contents are stored in either unstructured or semi-structured format due to which distillation of knowledge from this huge repository is a challenging task. In this paper, we propose a text mining approach to mine product features, opinions and their reliability scores from Web opinion sources. A rule-based system is implemented, which applies linguistic and semantic analysis of texts to mine feature-opinion pairs that have sentence-level co-occurrence in review documents. The extracted feature-opinion pairs and source documents are modeled using a bipartite graph structure. Considering feature-opinion pairs as hubs and source documents as authorities, Hyperlink-Induced Topic Search (HITS) algorithm is applied to generate reliability score for each feature-opinion pair with respect to the underlying corpus. The efficacy of the proposed system is established through experimentation over customer reviews on different models of electronic products.

## Categories and Subject Descriptors

I.5.4 [Pattern Recognition]: Applications—*Text processing*; I.7.5 [Document and Text Processing]: Document Capture—*Document analysis*

## General Terms

Algorithms, Design

## Keywords

Text mining; Opinion mining; Feature identification; Opinion reliability

\*To whom correspondence should be made

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WIMS '12, June 13-15, 2012 Craiova, Romania

Copyright © 2012 ACM 978-1-4503-0915-8/12/06 ...\$10.00.

## 1. INTRODUCTION

Due to increasing popularity of the Web 2.0 and easy access, it is emerging as a new medium, which describes individual experiences. Numerous merchant sites, forums, discussion groups and blogs are in existence that attract individual users to participate more actively to share their experiences. Consequently, a vast amount of opinion data is being generated by the Web users, which is symbolized by the new term “*user-generated contents*”. The online merchant sites provide space for customers to share experiences (opinions) about their products and consequently a number of opinion resources (a.k.a. review documents) exists for each product. The opinion resources play an important role both for the customers as well as for the manufacturers. The opinion sources are useful for the customers in choosing a right product based on the experiences of the existing users, whereas, they help product manufacturers to know the strength and weaknesses of their products from the perspective of end users. On one hand, the strengths could be used by the manufacturer for attracting potential customers, whereas on the other hand the weaknesses could be tackled in future versions of the product for customer satisfaction and retention in this competitive age [13].

Since customer feedbacks influence other customer's decision, the review documents have become an important source of information for business organizations to take it into account while developing marketing and product development plans. As the number of reviews that a product receives may grow rapidly and many times the reviews may also be quite lengthy, it is hard for the customers to analyze them through manual reading to make an informed decision to purchase a 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 read a few reviews in order to form a decision regarding the product and he/she may get only a biased view of the product. Similarly, manufacturers want to read the reviews to identify what elements of a product affect sales most, and a large number of reviews makes it hard for product manufacturers or business organizations to keep track of customer's opinions and sentiments on their products and services. Since, most of the reviews are stored either in unstructured or semi-structured format, the distillation of knowledge from this huge repository becomes a challenging task. It would be a great help for both customers and manufacturers if the reviews could be processed automatically and presented in a summarized

form highlighting the product features and users opinions expressed over them. In this paper, we propose a text mining approach to mine product features and opinions from review documents. As observed in [11], most product features can be found by exploiting local information and their Parts-Of-Speech (POS). Therefore, the proposed approach implements the information component extraction mechanism as a rule-based system, which applies both linguistic and semantic analysis on review documents. An information component is defined as a triplet  $\langle f, m, o \rangle$ , where  $f$  represents a feature generally identified as a noun phrase,  $o$  represents an opinion expressed over  $f$  generally identified as adjective, and  $m$  is a modifier generally used to model the degree of expressiveness of  $o$ . Since, for a product feature, different users may express same or different opinions and a single user (a review document) may express opinions on different features; in this scenario, a simple frequency-based summarization of the extracted feature-opinion pairs is not suffice to express their reliability. We have modeled the extracted feature-opinion pairs and resource documents as a bipartite graph. Considering feature-opinion pairs as hubs and documents as authorities in the bipartite graph, HITS [7] algorithm is applied to generate *reliability score* for each feature-opinion pair with respect to the underlying corpus. The *reliability score* is generated as a normalization of the hub score values.

The remaining paper is structured as follows. Section 2 presents a brief review of the existing opinion mining systems. Section 3 presents architecture and functional detail of the proposed 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.

## 2. RELATED WORK

Opinion mining generally refers to the process of extracting product features and opinions from review documents and summarizing them using a graphical representation. In recent past, a lot of works have been done in this area [5, 3, 20, 8, 17]. The existing approaches attempted to mine opinions at different levels of granularities including documents [18], sentences [6] and words [4]. In [18], Turney proposed an unsupervised learning algorithm to classify a review as *recommended* or *not recommended*, which applies POS analysis to identify opinion phrases in review documents and uses PMI-IR algorithm [19] to identify their semantic orientations. Generally, document-level opinion mining systems fail to reveal the product features liked or disliked by the users, rather they classify the reviews as positive or negative. A positive review does not mean that the opinion holder has positive opinion on all aspects or features of the product. Similarly, a negative review does not mean that the opinion holder dislikes every thing about the product. Keeping in mind the above facts, feature-based opinion mining is proposed in [5, 11, 14]. In [5], Hu and Liu have applied a three-step process for opinion mining. Starting with the identification of product features commented by the end users, they located the opinion sentences where features are commented and marked them either as positive or negative. Finally, the documents are summarized around each feature by classifying positive opinion sentences in one set and that the negative opinion sentences in another set. In [11], the authors have proposed a supervised pattern min-

ing method, which identifies product features from pros and cons sections of the review documents in an automatic way. In [14], the design of OPINE system based on an unsupervised pattern mining approach is presented, which extracts explicit product features using feature assessor and web PMI statistics. In [8], the authors have proposed a pattern mining method in which patterns are described as a relationship between feature and opinion pairs. Pattern mining technique is used to extract patterns and statistics from the corpus are used to determine the confidence score of the extraction. In [16], double propagation approach is used to extract opinion words and features using a seed opinion lexicon, and thereafter the newly extracted opinions and features are exploited for further opinion and feature extraction. Since a complete opinion is always expressed in one sentence along with its relevant feature [9], the feature and opinion pair extraction can be performed at sentence-level to avoid their false associations.

## 3. PROPOSED OPINION MINING SYSTEM

In this section, we present the architecture and functional detail of the proposed opinion mining system to identify feature-opinion pairs and their reliability scores with respect to an underlying corpus. Figure 1 presents the complete architecture of the proposed opinion mining system, which consists of five different functional components – *review documents crawler*, *document pre-processor*, *document parser*, *feature and opinion learner*, and *reliability score generator*. Further details about these modules are presented in the following sub-sections.

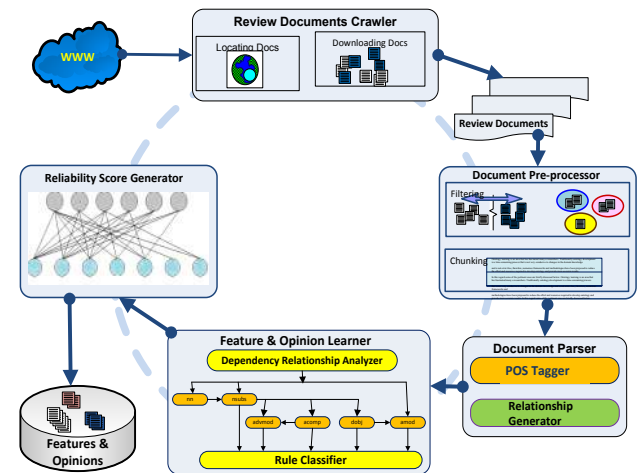


Figure 1: Architecture of the proposed opinion mining system

### 3.1 Review Documents Crawler and Document Pre-processor

For a target review site, the crawler retrieves review documents and stores them locally after filtering markup language tags. The filtered review documents are divided into manageable record-size chunks whose boundaries are decided heuristically based on the presence of special characters. It has been found that granularity of words, word

stems, and word synonyms may cause problem while extracting real features and opinion. We have applied rigorous preprocessing on review documents to filter out noisy reviews that are introduced either without any purpose or to increase/decrease the popularity of the product.

### 3.2 Document Parser

The functionality of this module is to facilitate the linguistic and semantic analysis of text for information component extraction. This module accepts record-size chunks generated by *document pre-processor* as input to assign Parts-Of-Speech (POS) tags to each word. It also converts each sentence into a set of dependency relations between the pair of words. For POS analysis and dependency relation generation purpose, we have used Stanford parser<sup>1</sup>, which is a statistical parser. As observed in [1], noun phrases generally correspond to product features, adjectives refer to opinions and adverbs are generally used as modifiers to represent the degree of expressiveness of opinions, we have applied POS-based filtering mechanism to avoid unwanted texts from further processing.

### 3.3 Feature and Opinion Learner

This module is responsible to analyze dependency relations generated by document parser and generate all possible information components from them. The dependency relations between a pair of words  $w_1$  and  $w_2$  is represented as  $relation\_type(w_1, w_2)$ , in which  $w_1$  is called head or governor and  $w_2$  is called dependent or modifier. The relationship  $relation\_type$  between  $w_1$  and  $w_2$  can be of two types – i) direct and ii) indirect [15]. In a direct relationship, one word depends on the other or both of them depend on a third word directly, whereas in an indirect relationship one word depends on the other through other words or both of them depend on a third word indirectly.

In line with [1], an information component is defined as a triplet  $\langle f, m, o \rangle$ , where  $f$  represents a feature generally expressed as a noun phrase,  $o$  refers to opinion which is generally expressed as adjective, and  $m$  is an adverb that acts as a modifier to represent the degree of expressiveness of the opinion. As pointed out in [16], opinion words and features are generally associated with each other and consequently, there exist inherent as well as semantic relations between them. Therefore, the *feature and opinion learner* module is implemented as a rule-based system, which analyzes the dependency relations to identify information components from review documents. For example, consider the following opinion sentences related to Nokia N95:

- (i) The screen is very attractive and bright.
- (ii) The sound some times comes out very clear.
- (iii) Nokia N95 has a pretty screen.
- (iv) Yes, the push email is the “Best” in the business.

In example (i), the *screen* is a noun phrase which represents a feature of *Nokia N95*, and the adjective word *attractive* can be extracted using nominal subject *nsubj* relation (a dependency relationship type used by Stanford parser) as an opinion. Further, using *advmod* relation the adverb *very* can be identified as a modifier to represent the degree

of expressiveness of the opinion word *attractive*. In example (ii), the noun *sound* is a nominal subject of the verb *comes*, and the adjective word *clear* is adjectival complement of it. Therefore, *clear* can be extracted as opinion word for the feature *sound*. In example (iii), the adjective *pretty* is parsed as directly depending on the noun *screen* through *amod* relationship. If *pretty* is identified as an opinion word, then the word *screen* can be extracted as a feature; like wise, if *screen* is identified as a feature, the adjective word *pretty* can be extracted as an opinion. Similarly in example (iv), the noun *email* is a nominal subject of the verb *is*, and the word *Best* is direct object of it. Therefore, *Best* can be identified as opinion word for the feature word *email*.

Based on these and other observations, we have defined six different rules to tackle different types of sentence structures to identify information components embedded within them. A summarized representation of these rules is presented in the following paragraphs.

**Rule-1:** In a dependency relation  $R$ , if there exist relationships  $nn(w_1, w_2)$  and  $nsubj(w_3, w_1)$  such that  $POS(w_1) = POS(w_2) = NN^*$ ,  $POS(w_3) = JJ^*$  and  $w_1, w_2$  are not stop-words, or if there exists a relationship  $nsubj(w_3, w_4)$  such that  $POS(w_3) = JJ^*$ ,  $POS(w_4) = NN^*$  and  $w_3, w_4$  are not stop-words, then either  $(w_1, w_2)$  or  $w_4$  is considered as a feature and  $w_3$  as an opinion.

**Rule-2:** In a dependency relation  $R$ , if there exist relationships  $nn(w_1, w_2)$  and  $nsubj(w_3, w_1)$  such that  $POS(w_1) = POS(w_2) = NN^*$ ,  $POS(w_3) = JJ^*$  and  $w_1, w_2$  are not stop-words, or if there exists a relationship  $nsubj(w_3, w_4)$  such that  $POS(w_3) = JJ^*$ ,  $POS(w_4) = NN^*$  and  $w_3, w_4$  are not stop-words, then either  $(w_1, w_2)$  or  $w_4$  is considered as the feature and  $w_3$  as an opinion. Thereafter, the relationship *advmod*( $w_3, w_5$ ) relating  $w_3$  with some adverbial word  $w_5$  is searched. In case of presence of *advmod* relationship, the information component is identified as  $\langle (w_1, w_2) \text{ or } w_4, w_5, w_3 \rangle$  otherwise  $\langle (w_1, w_2) \text{ or } w_4, -, w_3 \rangle$ .

**Rule-3:** In a dependency relation  $R$ , if there exist relationships  $nn(w_1, w_2)$  and  $nsubj(w_3, w_1)$  such that  $POS(w_1) = POS(w_2) = NN^*$ ,  $POS(w_3) = VB^*$  and  $w_1, w_2$  are not stop-words, or if there exist a relationship  $nsubj(w_3, w_4)$  such that  $POS(w_3) = VB^*$ ,  $POS(w_4) = NN^*$  and  $w_4$  is not a stop-word, then we search for *acompl*( $w_3, w_5$ ) relation. If *acompl* relationship exists such that  $POS(w_5) = JJ^*$  and  $w_5$  is not a stop-word then either  $(w_1, w_2)$  or  $w_4$  is assumed as the feature and  $w_5$  as an opinion. Thereafter, the modifier is searched and information component is generated in the same way as in Rule-2.

**Rule-4:** In a dependency relation  $R$ , if there exist relationships  $nn(w_1, w_2)$  and  $nsubj(w_3, w_1)$  such that  $POS(w_1) = POS(w_2) = NN^*$ ,  $POS(w_3) = VB^*$  and  $w_1, w_2$  are not stop-words, or if there exists a relationship  $nsubj(w_3, w_4)$  such that  $POS(w_3) = VB^*$ ,  $POS(w_4) = NN^*$  and  $w_4$  is not a stop-word, then we search for *doobj*( $w_3, w_5$ ) relation. If there exists a *doobj* relationship such that  $POS(w_5) = NN^*$  and  $w_5$  is not a stop-word then either  $(w_1, w_2)$  or  $w_4$  is considered as the feature and  $w_5$  as an opinion.

**Rule-5:** In a dependency relation  $R$ , if there exists a *amod*( $w_1, w_2$ ) relation such that  $POS(w_1) = NN^*$ ,  $POS(w_2) = JJ^*$ , and  $w_1$  and  $w_2$  are not stop-words then  $w_2$  is assumed to be an opinion and  $w_1$  as a feature.

**Rule - 6:** In a dependency relation  $R$ , If there ex-

<sup>1</sup><http://nlp.stanford.edu/software/lex-parser.shtml>

ist relationships  $nn(w_1, w_2)$  and  $nsubj(w_3, w_1)$  such that  $POS(w_1) = POS(w_2) = NN^*$ ,  $POS(w_3) = VB^*$  and  $w_1, w_2$  are not stop-words, or if there exists a relationship  $nsubj(w_3, w_4)$  such that  $POS(w_3) = VB^*$ ,  $POS(w_4) = NN^*$  and  $w_4$  is not a stop-word, then we search for  $dobj(w_3, w_5)$  relation. If  $dobj$  relationship exists such that  $POS(w_5) = NN^*$  and  $w_5$  is not a stop-word then either  $(w_1, w_2)$  or  $w_4$  is assumed as the feature and  $w_5$  as an opinion. Thereafter, the relationship  $amod(w_5, w_6)$  is searched. In case of presence of  $amod$  relationship, if  $POS(w_6) = JJ^*$  and  $w_6$  is not a stop-word, then the information component is identified as  $< (w_1, w_2)$  or  $w_4, w_5, w_6 >$  otherwise  $< (w_1, w_2)$  or  $w_4, w_5, - >$ .

### 3.4 Reliability Score Generator

During the information component extraction phase, a large number of noun, verb and adjectives are extracted that are not relevant to our task. Sometimes, it is observed that verbs are considered as noun due to parsing error. The basic reason for occurrence of these noises is the presence of ordinary nouns, verbs and adjective that are not actual features and opinions, but extracted as features and opinions due to parsing errors and their association with each other. Therefore, post processing is required as we are interested to find only those features on which customers express their opinions frequently. Consequently, for every relevant noun phrase (representing product feature) the list of all opinions and modifiers are compiled and stored. Similarly, if an adjective is related to various features, it is extracted as relevant opinion. Therefore, for every extracted real opinion word, the list of all features are compiled and stored in a structured form. Another issue is that, very often several customers comment on the same product feature and in many cases their opinions contradict. We handle the parsing errors and other noises as well as the contradiction of user comments by assigning a *reliability score*,  $0 \leq r \leq 1$ , to each *feature-opinion* pair. A higher score value for a pair reflects a tight integrity of the two components in a pair. For this, we follow the opinion retrieval model used by Li *et al.*[9] which is based on the HITS algorithm [7]. The extracted feature-opinion pairs are represented as an undirected bipartite graph based model which is then treated by the HITS algorithm [7] to generate *reliability scores* for *feature-opinion* pairs.

HITS algorithm distinguishes hubs and authorities in the set of objects. A hub object has links to many good authorities, and an authority object has high-quality content and there are many hubs linking to it. The hub and authority scores are computed in an iterative way. In this work, we have considered *feature-opinion* word pairs as hubs and review documents as authorities. Figure 2 shows an exemplar undirected bipartite graph in which hubs and authorities appear at upper and lower layers, respectively.

Formally, a bipartite graph is represented as a triplet of the form  $G = \langle V_p, V_d, E_{dp} \rangle$ , where  $V_p = p_{ij}$  is the set of feature-opinion pairs that have co-occurrence at sentence-level,  $V_d = d_k$  is the set of review documents containing feature-opinion pairs, and  $E_{dp} = \{e_{ij}^k | p_{ij} \in V_p, d_k \in V_d\}$  refers to the correlation between documents and feature-opinion word pairs. Each edge  $e_{ij}^k$  is associated with a weight  $W_{ij}^k \in [0, 1]$  to represent the strength or integrity of a relationship between the pair  $p_{ij}$  and the document  $d_k$ . The weight of a word pair  $p_{ij}$  in all sentences of the document  $d_k$

is calculated using equations 1, 2, 3, and 4, where  $|d_k|$  is the number of sentences in document  $d_k$  and  $0 \leq \alpha \leq 1$  is used as a trade-off parameter. The feature score is calculated using term frequency ( $tf$ ) and inverse sentence frequency ( $isf$ ) in each sentence of the document.  $tf(f_i, s_l)$  is the number of times  $f_i$  occurs in  $s_l$  sentence.  $N$  is the total number of sentences in the document and  $sf(f_i)$  is the number of sentences where the feature  $f_i$  appears [9, 10, 12]. Also,  $tf(o_j, s_l)$  is the number of times opinion  $o_j$  appears in a sentence  $s_l$  and  $asl$  is the average number of sentences in the document  $d_k$  [2].

$$W_{ij}^k = \frac{1}{|d_k|} \sum_{p_{ij} \in s_l \in d_k} [\alpha \times \text{fScore}(f_i, s_l) + (1 - \alpha) \times \text{oScore}(o_j, s_l)] \quad (1)$$

$$\text{fScore}(f_i, s_l) = tf(f_i, s_l) \times isf(f_i) \quad (2)$$

$$isf(f_i) = \log \left( \frac{N + 1}{0.5 \times sf(f_i)} \right) \quad (3)$$

$$\text{oScore}(o_j, s_l) = \frac{tf(o_j, s_l)}{tf(o_j, s_l) + 0.5 + \left( 1.5 \times \frac{\text{len}(s_l)}{asl} \right)} \quad (4)$$

The authority score  $AS^{(t+1)}(d_k)$  of document  $d_k$  and hub score  $HS^{(t+1)}(p_{ij})$  of  $p_{ij}$  at the  $(t + 1)^{th}$  iteration are computed based on the hub scores and authority scores obtained at the  $t^{th}$  iteration by using equations 5 and 6.

$$AS^{(t+1)}(d_k) = \sum_{p_{ij} \in V_p} W_{ij}^k \times HS^{(t)}(p_{ij}) \quad (5)$$

$$HS^{(t+1)}(p_{ij}) = \sum_{d_k \in V_d} W_{ij}^k \times AS^{(t)}(d_k) \quad (6)$$

The bipartite graph is represented in its adjacency matrix as follows:

$$L = (L_{i,j})_{|V_p| \times |V_d|} \quad (7)$$

such that:

$$\vec{a}^{(t+1)} = L \vec{h}^{(t)} \quad (8)$$

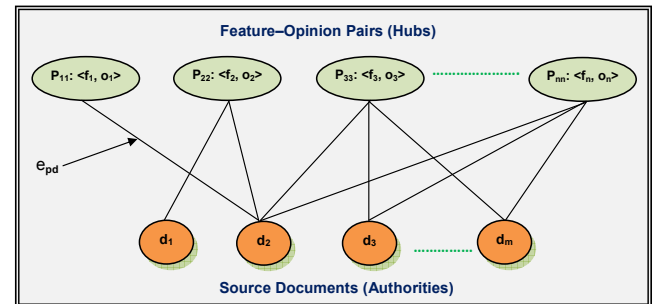


Figure 2: Bipartite link graph

and

$$\vec{h}^{(t+1)} = L^{(t)} \vec{a}^{(t)} \quad (9)$$

where,  $\vec{a}^{(t)} = [AS^{(t)}(d_k)]_{|V_d| \times 1}$  is the vector of authority scores for documents at the  $t^{th}$  iteration, and  $\vec{h}^{(t)} = [HS^{(t)}(p_{ij})]_{|V_p| \times 1}$  is the vector of hub scores for the feature-opinion pairs at  $t^{th}$  iteration.

For numerical computation of the final score, initial scores of all documents as well as feature-opinion word pairs are set to 1 and the above iterative steps are used to compute the new scores until convergence [10]. Usually the convergence of the above iteration algorithm is reached when the scores computed at two successive iterations for any feature-opinion word pair or review document falls below a given threshold. In our experiment the threshold value is set to 0.0001. After final convergence of iterations in HITS algorithm, the generated hub scores of  $(f_i, o_j)$  pairs present soundness of the integration of feature and opinion in the pair with respect to the documents where they occur. We calculate the *reliability score*,  $r_{ij}$ , for pair  $(f_i, o_j)$  by normalizing this score using *min-max* normalization to scale it in  $[0, 1]$  as shown in equation 10, where  $HS_{(p_{ij})}^n$  denotes hub score of  $p_{ij}$  after  $n$ th iteration (after convergence) and *NewMax* and *NewMin* values are set to 1 and 0 respectively. This metric determines the reliability of an opinion expressed over a product feature.

$$r_{ij} = \frac{HS_{(p_{ij})}^n - \min_{xy} \{HS_{(p_{xy})}^n\}}{\max_{xy} \{HS_{(p_{xy})}^n\} - \min_{xy} \{HS_{(p_{xy})}^n\}} \times (\text{NewMax} - \text{NewMin}) + \text{NewMin} \quad (10)$$

## 4. EXPERIMENTAL RESULTS AND EVALUATION

In this section, we present the experimental setup and evaluation results of the proposed opinion mining system. The data samples used in our experimental work consist of 400 review documents on different models of cell phone crawled from [www.amazon.com](http://www.amazon.com). In our implementation, the dataset is crawled using *crawler4j* API<sup>2</sup> which are then pre-processed by some filtering to smoothen the noise and chunking to decompose the text into individual meaningful chunks. Using *Stanford Parser* API<sup>3</sup> the text chunks are further broken down to separate the different parts of speech (POS). Our rule-based system described in section 3.3 is applied to mine features and opinions along with the modifiers (if present). Initially, a total of 4333 noun (or verb) and adjective pairs were extracted by the system, out of which 1366 candidate features were retained after filtering for further analysis. Upon observations, we found that occurrence frequencies of real features are very high in review documents. This is due to the tendency that different reviewers refer same features with different comment words to express different opinions. After collecting the *feature-opinion* pairs, *reliability score* is calculated for each of them by the Java based *reliability score generator* as described in section 3.4. These scores assist in feasibility study of the pairs. Table 1 presents a partial list of feasible features along with their opinions and modifiers.

<sup>2</sup><http://code.google.com/p/crawler4j/>

<sup>3</sup><http://nlp.stanford.edu/software/lex-parser.shtml>

**Table 1: A partial list of extracted features, opinions and modifiers**

Feature	Modifier	Opinion
player	enough, very	good, nice
camera	much, very, pretty	worse, easy, great, good
screen	pretty, barely, fairly, very	solid, visible, responsive, receptive, good
software	rather, definitely, not	easy, slow, flimsy, limited, seamless

Evaluation of the experimental results is performed using standard Information Retrieval (IR) metrics *Precision*, *Recall* and *F-score* that are defined in equations 11, 12, and 13, respectively. In these equations, TP indicates true positive, which is defined as the number of *feature-opinion* pairs that the system identifies correctly, FP indicates false positive which is defined as the number of *feature-opinion* pairs that are identified falsely by the system, and FN indicates false negatives which is the number of *feature-opinion* pairs that the system fails to identify.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (11)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (12)$$

$$F\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (13)$$

### 4.1 Evaluating Feature and Opinion Learner

To the best of our knowledge, no benchmark data is available in which features and opinions are marked for electronic products, therefore we performed a manual evaluation to judge the overall performance of the system. Features and their corresponding opinions were extracted from review documents by the *Feature and Opinion Learner* component of the proposed system. Parallely, we collected all the feature and opinion pairs manually from these documents. Thereafter, comparing the two sets of pairs TP, FP and FN are calculated. Macro-averaged performance is obtained to present a synthetic measure of performance by simply averaging the result. The total count obtained for TP, FP, and FN are 641, 167, and 266, respectively. On the basis of these count, the *Precision*, *Recall* and *F-score* values of the system are found to be 79.33%, 70.67% and 74.75%, respectively. We have observed that, direct and strong relationship between words causes extraction of those nouns (or, verbs) and adjectives that are not relevant feature-opinion pairs. As a result, counts for FP (false positive) get increases which has an adverse effect on the value of precision. Since most of the reviewers use informal approach while commenting, reviews are generally lack in grammatical correctness and pose a number of challenges for natural language parser, the recall value obtained is lower than precision which is an indication of system inability to extract certain feature-opinion pairs correctly.

Table 2 presents the result summary found for review documents on four different products, Nokia, AT&T, LG and BlackBerry cellphones. For 10 documents from each category, we have presented the true positives, false positives

**Table 2: Misclassification matrix for four different cellphone models**

Product	TP	FP	FN	Precision (%)	Recall (%)	F-Score (%)
Nokia Cellphone	74	37	29	66.67	71.84	69.15
AT&T Cellphone	70	25	26	73.68	72.91	73.29
LG Cellphone	51	31	33	62.19	60.71	61.44
BlackBerry Cellphone	42	16	17	72.41	71.18	71.78
Macro-Average	237	109	105	68.50	69.30	68.90

and false negatives individually, which are used to calculate *Precision*, *Recall* and *F-scores*. On these 40 documents, we found the macro-average *Precision*, *Recall* and *F-score* as 68.73%, 69.16% and 68.91% respectively. The data in table 2 is visualized in figure 3 to present a comparative view of the accuracy values for four different models of cellphone. It can be observed that for each metric the values do not undergo much variation which shows the applicability of the proposed method irrespective to the domain of review documents.

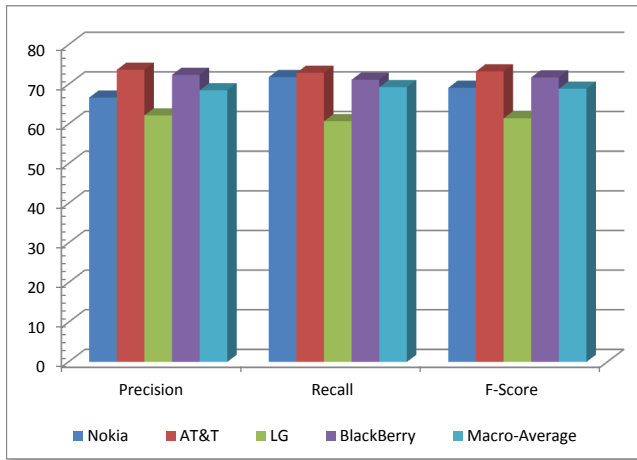


Figure 3: A comparison of *Precision*, *Recall* and *F-score* values for different cellphone models

## 4.2 Evaluating Reliability Score Generator

In previous subsection, we evaluated the quality of *feature-opinion* pairs generated by our rule-based *Feature and Opinion Learner* in terms of *Precision*, *Recall* and *F-score*. The next component in the proposed system captures further information about each extracted pair in the form of a value ranging between 0 and 1, to assist a viewer in determining the *reliability* of the pair.

In table 3, we present top 5 hub scores and their final reliability scores assigned to the most reliable feature and opinion pairs in our experiment. The highest *reliability score* for the *phone-thin* pair indicates *thin* as the most prominent quality opined by the reviewers. In second position we find *phone-feature* pair with a *reliability score* of 0.9936, and as obvious this pair is incorrectly extracted by our rule-based model. In the top 5 positions, we can see that all of them are about the product *phone* which indicates that it is the most popular product among the reviewers which make them to comment on it. Table 4 presents hub and *reliability scores* for some randomly selected *feature-opinion* pairs. Similarly,

in table 5, we present the top 5 authority scores along with their normalized values assigned to review documents.

Table 3: Top-5 hub scored feature and opinion pairs and their reliability scores

Feature	Opinion	HS	Reliability Score (r)
Phone	Thin	5.7033	1.0000
Phone	Feature	5.6670	0.9936
Phone	Great	5.6562	0.9917
Phone	Good	5.6523	0.9911
Phone	Easy	5.5307	0.9697

Table 4: Some exemplar feature-opinion pairs along with their hub and reliability scores

Feature	Opinion	HS	Reliability Score (r)
Phone	Thin	5.7033	1.0000
OS	Tricky	2.2557	0.3955
Screen	Large	1.9605	0.3437
Feature	Unlock	1.8612	0.3263
Camera	MP	1.4684	0.2575
Quality	Good	1.3275	0.2328
Battery	Bad	1.2718	0.2230
Keyboard	Great	1.2379	0.2170
Problem	Indicator	1.1610	0.2036
Picture	Good	0.8909	0.1562

Table 5: Top-5 authority scored review documents

Authority Name	AS	Normalized AS
111LGVuCUC	18.8624	1.000
115LGVuCUC	13.2042	0.6772
260ATTTiltPhone	13.1702	0.6752
74NokiaNSma	12.7430	0.6509
59NokiaNSma	12.7119	0.6491

## 5. CONCLUSION AND FUTURE WORK

In this paper, we have presented an opinion mining system which implements a rule-based system to identify candidate feature-opinion pairs from review documents. Our system is able to identify product features and opinions that are related either directly or indirectly. The extracted feature-opinion pairs along with the source documents are modeled using a bipartite graph. The graph-based ranking algorithm HITS is applied on the bipartite graph for feasibility analysis and reliability score generation with respect to the underlying corpus. Currently, we are refining the rule set to identify more dependency relationships to improve the precision and recall values of the proposed system and to identify implicit



features. Handling informal texts that are very common with review documents is also one of our future works.

## 6. ACKNOWLEDGEMENTS

The authors acknowledge the support provided by King Abdulaziz City for Science and Technology (KACST) and King Saud University (KSU), Kingdom of Saudi Arabia. This work has been funded by the KACST under the NPST project number 11-INF1594-02.

## 7. REFERENCES

- [1] M. Abulaish, Jahiruddin, M. Doja, and T. Ahmad. Feature and opinion mining for customer review summarization. In *Proceedings of the 3rd International Conference on Pattern Recognition and Machine Intelligence, Springer-Verlag, Berlin, Heidelberg*, pages 219–224, 2009.
- [2] J. Allan, C. Wade, and A. Bolivar. Retrieval and novelty detection at the sentence level. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, Toronto, Canada*, 2003.
- [3] A. Andreevskaia and S. Bergler. Mining wordnet for fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, EACL '06*, pages 209–216, 2006.
- [4] E. Breck, Y. Choi, and C. C. Identifying expressions of opinion in context. In *Proceedings of the 20th international joint conference on Artificial intelligence, Menlo Park, CA, USA*, pages 2683–2688, 2007.
- [5] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '04*, pages 168–177. ACM, 2004.
- [6] S. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics, Geneva, Switzerland*, pages 1367–1373, 2004.
- [7] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46:604–632, 1999.
- [8] N. Kobayashi, K. Inui, and Y. Matsumoto. Extracting aspect-evaluation and aspect-of-relations in opinion mining. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague*, pages 1065–1074, 2007.
- [9] B. Li, L. Zhou, S. Feng, and K. Wong. A unified graph model for sentence-based opinion retrieval. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden*, pages 1367–1375, 2010.
- [10] F. Li, Y. Tang, M. Huang, and X. Zhu. Answering opinion questions with random walks on graphs. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Suntec, Singapore*, pages 737–745, 2009.
- [11] 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, Japan*, pages 342–351, 2005.
- [12] J. Otterbacher, G. Erkan, and D. Radev. Using random walks for question-focused sentence retrieval. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver*, pages 915–922, 2005.
- [13] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2:1–135, January 2008.
- [14] A. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, Canada*, pages 339–346, 2005.
- [15] G. Qiu, B. Liu, J. Bu, and C. Chen. Expanding domain sentiment lexicon through double propagation. In *Proceedings of the 21st international joint conference on Artificial intelligence, San Francisco, CA, USA*, pages 1199–1204, 2009.
- [16] G. Qiu, B. Liu, and C. Chen. Opinion word expansion and target extraction through double propagation. *Association for Computational Linguistics*, 37:9–27, 2010.
- [17] Q. Su, X. Xu, H. Guo, Z. Guo, X. Wu, B. Zhang, X. and Swen, and Z. Su. Hidden sentiment association in chinese web opinion mining. In *Proceeding of the 17th international conference on World Wide Web, Beijing, China*, pages 959–968, 2008.
- [18] 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, Philadelphia, Pennsylvania*, pages 417–424, 2002.
- [19] P. D. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the 12th European Conference on Machine Learning, EMCL '01*, pages 491–502. Springer-Verlag, 2001.
- [20] L. Zhuang, F. Jing, and X.-Y. Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management, CIKM '06*, pages 43–50. ACM, 2006.