Clustering Technique for Feature Segregation in Opinion Analysis

Tanvir Ahmad
Jamia Millia Islamia,
Department of Computer Engineering,
New Delhi, India.

ABSTRACT

The World Wide Web (WWW) is a reservoir of enormous amount of data which is primarily embedded within unstructured text documents. E-commerce websites, social networking sites, and discussion forums have become a common place for writing informal opinions about products and other related information. A substantial amount of research has been directed towards mining these texts and concludes on the overall meaning of the users and to assign a grade to the products under discussion. These grading systems often become helpful for users to get an informed opinion about the products he/she wants to buy. There have been different techniques adopted by the opinion website developers to provide end users an overall meaning of the contents, like numerical rating on some predefined scale, star rating, and calculation of the percentage of users who are satisfied or dissatisfied with a product. However, all these methods have failed to segregate the features on the basis of opinion expressed in them or to cluster them in different group which gives a general insight into the features grouped together. In this paper, a framework has been presented which first extracts the feature, modifier and opinion from the dataset and then using clustering mechanism divides them into discrete clusters on the basis of users' opinion, in which the intra-cluster similarity between the features are high whereas the inter-cluster similarity is very low.

General Terms
Opinion Mining, Natural Language Processing

Keywords
Pattern Recognition, Feature Extraction, Clustering Technique

1. INTRODUCTION

In recent past, due to increasing popularity of the World Wide Web (WWW) and online social media, including online social networking sites, micro-blogging sites, online discussion forums, newsgroups, review sites, blogs, etc. there has been an unabated growth in user-generated contents causing the problem of *information overload*. Though the amount of useful information and knowledge contained in such data sources is very high, the research challenge lies in the fact that distillation of knowledge from such repository is very difficult, as most of them are either unstructured or semi-structured in nature. As a result, there is an increasing need of converting the information embedded within unstructured or semi-structured sources into a structured form, generally termed as *database curation*, without which the knowledge cannot be assimilated in a meaningful way which can be perceived by the users without exploring the pile of documents. Once the data contained in unstructured or semi-structured text documents are mapped to a structured format, the novel data mining techniques, especially the frequent and sequential pattern mining techniques, can be applied to mine meaningful patterns from textual data.

Though, a number of techniques including document classification and clustering, information extraction, text summarization, etc. have been developed to analyze information contained in textual data, they are not sufficient to be applied on opinion data sources that have the intricacy of the embedded natural language in the documents. Though a good number of research efforts have been diverted towards analyzing opinion sources, including feature and opinion extraction, and sentiment analysis, to the best of our knowledge, no research effort has been made to identify patterns among product features based on the associated opinions expressed by the users and then to cluster it in such a way that the satisfaction/dissatisfaction of the user on a particular feature influence his/her satisfaction/dissatisfaction over other features by analyzing the set of features lying in a group. Moreover, this kind of clustering of feature patterns could be very useful for feature-based target marketing of products through highlighting high influential features at the outset to attract potential customers.

In this paper, the clustering task has been done using two main steps. In the first step, the *<f, m, o>* triplet has been extracted, where *f* stands for the features, *o* is opinion and *m* is the optional modifier expressed on that measure which is used to intensify or diminish the effect of the opinion. For example in a statement *"The quality of food in the hotel was extremely good"*, the extracted *f* will be *food*, *o* will be *good* and *m* will be *extremely*. The list of the extracted triplets are maintained in a structured data file so that further processing can be applied on it in order to perform the granular approach of feature analysis. In the second step the k-means clustering algorithm has been used on the extracted triplets by constructing a matrix of features and opinions. Each entry of the matrix will contain the number of opinions expressed on that feature. Thereafter the clustering algorithm is run using WEKA which makes clusters taking different values of *k.* The results of the clusters are interpreted and presented to the user.

In order to establish the efficacy of the proposed methods in this paper, the dataset related to the digital camera domain has been used in which review documents related to four different models of digital camera are considered.

2. RELATED WORKS

Classification of a document or sentence into positive, negative or neutral classes is not the only solution to many problems as many times the need arises to perform feature-based opinion mining so that the users get a more informed opinion on the features of a product. Moreover, a positive opinion of a document does not mean that the author likes all the features of the product, and similarly a negative opinion does not means that author dislikes all the features of the

product. For example, in a product review, the reviewers often write both positive and the negative opinion of the product, and may write a final opinion of the product at the end. In order to find the list of positive and negative features, it is required to go in details at the sentence level categorization. As given in [1], there are two tasks which have to be performed at this level.

- To identify and extract the features of a product on which the reviewers have expressed their opinions. For example, in the sentence *"the size of the display of the mobile phone is amazing"*, the product feature is *"display"*.
- To determine the polarity (positive, negative or neutral) of the opinions. For example, in the previous sentence, the polarity of the opinion word "*amazing"* expressed over the product feature "*display*" is positive.

An opinion can be expressed on any subject which can be a product, an organization, an individual, a topic, an event, a policy, or a news etc. As mentioned in [2], the general term "object" is used to denote entities that have been commented by the users. Each object has a set of components and also a set of attributes. For example, a product can have different sub-components, a topic can have different sub-topic, and so on. Similarly, in case of a mobile phone, the set of components may include lens, battery, camera, memory, display, weight, and size. Each of these components may have separate sub-components like a battery has *battery life, battery size, battery weight, battery type, e*tc.

Some of the algorithms used by the researchers to find patterns in opinionated texts are label sequential rules (LSR) as discussed in [3, 4]. In this method, features can be noun, adjective, verb or adverb, and the labels along with their POS tags used for mining are {$feature, NN}, {$feature, JJ}, {$feature, RB}, etc., where $feature denotes a feature extracted from the text, and NN, JJ, VB, and RB stands for Noun, Verb, Adjective, and Adverb, respectively. It has that 60-70% of the features are explicit noun phrases, a small portion of the explicit features are verbs and 20-30% of the features are implicit features. A word that indicates an implicit feature is called an implicit feature indicator.

In [16] a rule based system has been proposed to identify features. Some other notable works in this area has been done by Ding et al.[5], Bodendorf et al. [6], Chaoji et al. [7], Balahur et al. [8], Lafferty et al. [9], and Freitag et al. [10].

3. PROPOSED OPINION ANALYSIS SYSTEM

In this section, the design of the clustering techniques has been proposed for feature segregation from the point of the view of opinion mining in figure 1. The proposed design consists of the following key functionalities – *Document Pre-processing and Parsing, Subjective/Objective Analyzer, Document Parser, Feature and Opinion Learner and Pattern Characterization and Cluster Analysis.* Further details of the functionalities are presented in the following sub-sections.

3.1 Document Pre-Processor

This module is responsible to pre-process the review documents by identifying relevant portions of a text document. This module consists of a Markup Language (ML) tag filter, which divides an unstructured web document into individual record-size chunks, cleans them by removing ML

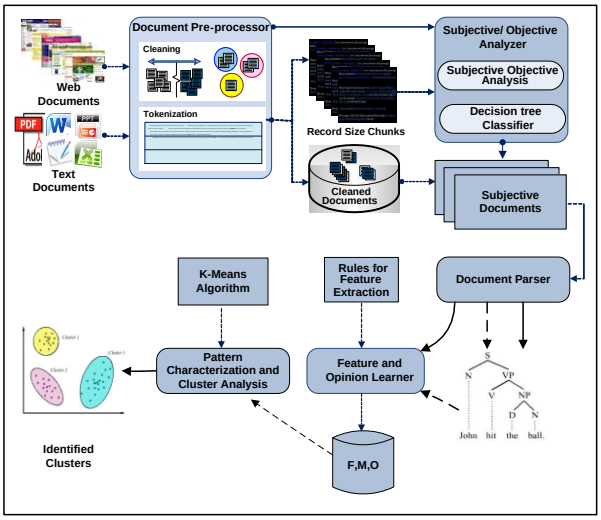

Figure 1: Architecture of the proposed Clustering Technique for Feature Clustering System

tags, and presents them as individual unstructured record documents for further processing. The cleaned documents are converted into numeric vectors using unigram model for the purpose of subjectivity/objectivity analysis. In document vectors a value represents the likelihood of each word being in a subjective or objective sentence.

3.2 Subjective/Objective Analyzer

As stated by Pang and Lee [11] subjective sentences are expressive of the reviewer's sentiment about the product, and objective sentences do not have any direct or obvious bearing on or support of that sentiment. Therefore, the idea of subjectivity analysis is used to retain segments (sentences) of a review that are more subjective in nature and filter out those that are more objective. This increases the system performance both in terms of efficiency and accuracy. The idea proposed by Yeh in [12] is used to divide the reviews into subjective parts and objective parts. In this paper he expressed the idea of cohesiveness to indicate segments of a review that are more subjective in nature versus those that are more objective. The experiment used a corpus of subjective and objective sentences used in [11] for training purpose. The training set is used to get the probability for each word to be subjective or objective, and the probability of a sentence to be subjective or objective is calculated using the unigram model. The Decision Tree classifier of Weka [13] is trained to classify the unseen review sentences into subjective and objective classes

3.3 Document Parser

Since our aim is to extract product features from text documents, all subjective sentences are parsed using Stanford Parser , which assigns Parts-Of-Speech (POS) tags to English words based on the context in which they appear. The POS information is used to locate different types of information of interest inside the text documents. For example, generally noun phrases correspond to product features therefore the first step is to extract nouns from the documents in order to generate frequent patterns. Each sentence is converted into dependency tree using Stanford Parser. The dependency tree, also known as word-word relationship, encodes the grammatical relations between every pair of words. The Stanford parser gives the output in the form of a dependency tree as well as typed dependency. The typed dependency diagram will be ultimately used for the extraction of the features after applying the rules.

3.4 Feature and Opinion Learner

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

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

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

3.4.1 Information Component Extraction

The information component extraction mechanism is implemented as a rule-based system [16] which analyses dependency tree to extract information components. The rules are presented below to highlight the function of the system.

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

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

Rule 3: In a dependency tree T , if there exists a *amod*(w_i, w_j) relation such that POS(w_j) ≠ NN* or POS(w_j) ≠ DET*, w_i and w_j are not stop-words and the sentence does not contain any *subj* relation then extract (w_i , w_j) and w_i is assumed to a feature and w_j to be the opinion.

3.4.2 Feature and Opinion Extraction

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

The *tf-idf* value for each noun phrase is calculated using equations 3.1 and 3.2 where, $tf(t_i)$ is the number of documents containing t_i, |D| is the total number of documents and $\left|\{d_j : t_i \in d_j\}\right|$ is the number of documents where t_i appears. All those noun phrases having *tf-idf* value above a threshold are considered as relevant features. Thereafter, for each retained feature, the list of opinion words and modifiers are compiled from information components and are stored in a structured form.

$$tf - idf(t_i) = tf(t_i) \times idf(t_i) \quad (1)$$

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

This module uses a corpus of 1025[1] customer reviews on four different models of *digital camera* in order to extract the feature and their related information.

From the dataset, a total of 182 features were extracted from the documents and the top 10 features were retained after feasibility analysis. The list of top 10 features extracted from the dataset after applying the rules in information component extraction has been shown in table 1. For each document a list of features found in that document was compiled and stored in a data file. The modifier and their associated opinions were not taken into consideration for the purpose of frequent pattern mining as they will be used in the next chapter for calculation of weights of the features and their clustering.

Table 1: List of top-10 features extracted from documents

Feature Number	Features
F1	Zoom
F2	LCD
F3	Picture
F4	Lens
F5	Photos
F6	Battery
F7	Price
F8	Camera
F9	Size
F10	Picture Quality

The performance of the whole system has been presented which is analyzed by taking into account the performance of the *feature and opinion* extraction process. Since terminology and complex proper names are not found in Dictionaries, an obvious problem of any automatic method for concept extraction is to provide objective performance evaluation. Therefore manual evaluation has been performed to judge the overall performance of the system. For evaluation of the experimental results, the standard IR performance measures have been taken. From the extraction results, true positive *TP* (number of correct feature-opinion pairs the system identifies

[1] Reviews were taken from www.ebay.com and some other websites

as correct), the false positive *FP* (number of incorrect feature-opinion pairs the system falsely identifies as correct), true negative *TN* (number of incorrect feature-opinion pairs the system identifies as incorrect), and the false negatives *FN* (number of correct feature-opinion pairs the system fails to identify as correct) has been calculated. Table 2 shows the result of the various measures achieved on the dataset.

Table 2: Various measures achieved on the dataset

Product Name		TP	FP	FN	TN	Precision (%)	Recall (%)	F1-measure (%)	Accuracy
Digital Camera	Canon	124	17	82	341	87.94	60.19	71.47	82.45
	Kodak	78	10	68	185	88.64	53.42	66.67	77.13
	Nikon	176	16	132	425	91.67	57.14	70.40	80.24
	Panas-onic	147	16	72	314	90.18	67.12	76.96	83.97
Macro-Average						**89.61**	**59.47**	**71.37**	**80.95**

3.5 Pattern Characterization and Cluster Analysis

Feature clustering involves a method to cluster the extracted features into various groups such that the intra-group similarity is high but the inter-group similarity is low. The objective behind feature clustering is to identify the features that are more or less similar based on the opinions expressed over them. On analysis, it was found that the nature of the graph makes it most successful method for applying clustering algorithms [14]. The task is to identify the features that are very much related together and to eliminate unimportant features. An important objective from the manufactures' point of view to identify the group of features that are inter-related in such a way that improving one feature may indirectly improve the rank of other features. Once this task is performed with a reasonable accuracy, it can be deduced that the set of features belonging to a group form a cohesive group of features.

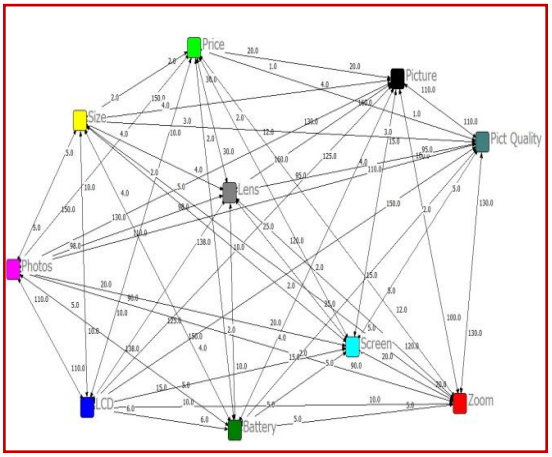

Figure 2: Network diagram of the top-10 features

In order to perform clustering, a network diagram of the extracted top-10 features has been drawn in order to know the relationship of the features among themselves. In order to draw the network diagram UCINET v6.0 for Windows tool [15] has been used whose output appears as a set of nodes and the interconnections between them. A particular feature having no interaction with other features is plotted as an isolated node. Figure 2 shows the network diagram of the top-10 features, in which the weights of the edges are shown alongside the edges.

Feature clustering has been done by identifying 25 different features, as shown in table 3, from the list of extracted information triplets. Along with the features, the list of opinions expressed over them has also been extracted, as shown in table 4.

For clustering, the *k*-Means algorithm has been implemented which is a component of WEKA machine learning tool. To execute this algorithm, it is required to convert the extracted features and opinions triplets into the form which can be executed by the tool. Therefore, the data file is converted into the Attribute Relationship File Format (*.arff*), which requires the attribute as well as the data embedded in the same file. From the dataset, the occurrence of the opinions for each feature and consequently generated a matrix of order 25 x 20 which contains normalized numeric values reflecting feature-opinion associations has been calculated. A snapshot of the data file generated from digital camera documents for clustering is shown in figure 3, and the corresponding .arff file with the dataset loaded for generating clusters is shown in figure 4. Figure 5 shows the output of the WEKA program with k = 4 clusters. The number of instances falling is each clusters can be seen at the bottom of the figure.

Table 3: Top-25 features extracted from the list of identified information triplets

Sl. No.	Identified Features	Sl. No.	Identified Features
1.	Camera	14.	Resolution
2.	Picture	15.	Flash
3.	Lens	16.	Viewing Screen
4.	Zoom	17.	Optics
5.	Photos	18.	Button
6.	LCD	19.	Processing time
7.	Battery	20.	Software
8.	Price	21.	Functions
9.	Size	22.	Clarity
10.	Video	23.	Share Software
11.	Color	24.	Money
12.	Weight	25.	Wide Angle
13.	Display		

Table 4: Top-20 opinions extracted from the list of identified information triplets

Sl.No.	Opinion extracted (O_i)	Sl. No.	Opinion extracted (O_i)
1.	Amazing	11.	Outstanding
2.	Awesome	12.	Easy
3.	Very good	13.	Large
4.	Great	14.	Fast
5.	Excellent	15.	Affordable
6.	Impressive	16.	Superior
7.	Nice	17.	Disappointing
8.	Perfect	18.	Slow
9.	Descent	19.	Bad
10.	Fantastic	20.	Average

	Amazing	Awesome	V.good	Great	Excellent	Impressive	Nice	Perfect	Descent	Fantastic	Outstanding	Easy	Large	Fast	Affordable	Superior	Disappointing	Slow	Bad	Average
Camera	25	22	30	18	30	28	20	15	5	10	19	0	0	0	5	0	0	0	0	0
Picture	31	19	13	15	17	20	5	30	0	8	11	0	0	0	0	0	0	0	0	0
Lens	15	9	15	5	17	12	2	5	6	5	11	0	0	0	0	0	3	0	0	0
Zoom	8	12	10	16	16	15	3	12	5	7	5	0	0	0	0	0	15	0	0	0
Photos	26	15	20	22	8	8	7	10	9	7	10	0	0	0	0	0	4	0	0	0
LCD	8	9	5	11	17	16	7	20	5	2	17	0	0	0	0	0	0	0	0	0
Battery	0	0	2	3	0	1	1	2	1	1	2	0	0	0	0	0	20	0	8	20
Price	0	0	2	3	0	2	0	1	0	0	1	0	0	0	6	0	15	0	2	10
Size	6	1	0	2	2	1	8	18	15	10	1	0	0	0	0	0	0	0	0	0
Video	0	0	1	2	0	0	1	1	1	0	0	0	0	0	0	0	24	0	0	12
Color	2	3	0	12	5	2	0	0	8	6	2	0	0	0	0	10	0	0	0	0
Weight	0	1	0	2	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0
Display	4	2	8	2	8	16	12	10	5	5	8	0	18	0	0	0	0	0	0	0
Resolution	3	14	10	2	0	0	1	5	0	2	0	0	0	4	0	0	0	0	0	0
Shutter speed	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	6	8	4	0
View screen	0	0	2	2	7	0	1	0	0	0	0	0	16	0	0	0	0	0	0	0
Optics	2	1	9	3	15	5	2	0	0	0	0	3	2	0	5	0	0	0	0	0
Button design	0	0	5	0	5	8	9	6	0	0	0	5	0	6	0	0	0	0	0	0
Proc time	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	5	5
Software	0	0	0	2	1	1	0	0	0	0	0	0	0	0	0	2	7	2	0	15
Function	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	2	0	2	0
Clarity	5	6	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Share s/w	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	0
Money	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	5
Wide angle	0	0	0	2	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0

Figure 3: Data file generated for the digital camera documents for clustering

```
@relation Feature_Cluster

@attribute    amazing real
@attribute    awesome real
@attribute    very_good real
@attribute    great real
@attribute    excellent real
@attribute    impressive real
@attribute    nice real
@attribute    perfect real
@attribute    descent real
@attribute    fantastic real
@attribute    outstanding real
@attribute    easy real
@attribute    large real
@attribute    fast real
@attribute    affordable  real
@attribute    superior real
@attribute    disappointing real
@attribute    slow  real
@attribute    bad  real
@attribute    average  real

@data
0.110132159,0.0969163,0.13215859,0.079295154,0.13215859,0.123348018,0.088105727,0.06
6079295,0.022026432,0.044052863,0.083700441,0,0,0,0.022026432,0,0,0,0
0.183431953,0.112426036,0.076923077,0.088757396,0.100591716,0.118343195,0.029585799
,0.177514793,0,0.047337278,0.065088757,0,0,0,0,0,0,0,0,0
0.142857143,0.085714286,0.142857143,0.047619048,0.161904762,0.114285714,0.019047619
,0.047619048,0.057142857,0.047619048,0.104761905,0,0,0,0,0,0.028571429,0,0,0
0.064516129,0.096774194,0.080645161,0.129032258,0.129032258,0.120967742,0.024193548
,0.096774194,0.040322581,0.056451613,0.040322581,0,0,0,0,0,0.120967742,0,0,0
0.178082192,0.102739726,0.136986301,0.150684932,0.054794521,0.054794521,0.047945205
,0.068493151,0.061643836,0.047945205,0.068493151,0,0,0,0,0,0.02739726,0,0,0
0.068376068,0.076923077,0.042735043,0.094017094,0.145299145,0.136752137,0.05982906,
0.170940171,0.042735043,0.017094017,0.145299145,0,0,0,0,0,0,0,0,0
0,0,0.032786885,0.049180328,0,0.016393443,0.016393443,0.032786885,0.016393443,0.0163
93443,0.032786885,0,0,0,0,0,0.327868852,0,0.131147541,0.327868852
0,0,0.047619048,0.071428571,0,0.047619048,0,0,0.023809524,0,0,0.023809524,0,0,0,0.142857
143,0,0.357142857,0,0.047619048,0.238095238
0.09375,0.015625,0,0.03125,0.03125,0.015625,0.125,0.28125,0.234375,0.15625,0.015625,0,0,
0,0,0,0,0,0
```

Figure 4: Dataset converted to WEKA format (.arff) for clustering

3.5.1 Performance Evaluation

To evaluate the efficacy of the proposed clustering approach, the clustering results has been analysed for different values of k = 2, k = 3, and k = 4. To run the clustering algorithm, the Euclidean distance measures, which refers to distance between two points in a straight line has been used. The advantage of this distance measure lies in its fair measure of how similar ratings are for specific preferences or items. The other parameters like maximum number of iterations was set to 500 and since there were no missing values in the dataset, the parameter for *dontReplaceMissingValues* has been set as "true". The sample results for *k = 2*, *k = 3*, and *k = 4* are shown in table 5.

```
kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 21.245022913671658
Missing values globally replaced with mean/mode

Cluster centroids:
                          Cluster#
Attribute          Full Data          0          1          2          3
                      (25)        (2)        (4)       (11)        (8)
=============================================================================
amazing             0.0548     0.2033     0.0336          0     0.1037
awesome             0.0599     0.3707      0.009     0.0061     0.0816
very_good           0.0567      0.122     0.0667     0.0095     0.1005
great               0.0678     0.1577     0.0308     0.0785     0.1117
excellent           0.0674          0     0.1191     0.0061     0.1429
impressive          0.0523          0     0.0902     0.0119     0.1019
nice                0.0403     0.0122     0.1219     0.0166     0.0389
perfect             0.0917      0.061     0.1299      0.093     0.0784
descent             0.0284          0     0.0713     0.0037      0.048
fantastic           0.0261     0.0244     0.0518     0.0015     0.0476
outstanding         0.0281          0     0.0243     0.0051     0.0685
easy                0.0357          0     0.0284     0.0649      0.008
large               0.0319          0     0.1888          0     0.0053
fast                0.0134     0.0488     0.0341     0.0091          0
affordable          0.0109          0          0      0.013     0.0161
superior             0.008          0          0          0      0.025
disappointing       0.0981          0          0     0.2069     0.0221
slow                0.0547          0          0     0.1242          0
bad                 0.0837          0          0     0.1902          0
average             0.0703          0          0     0.1597          0



Time taken to build model (full training data) : 0.02 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       2 (  8%)
1       4 ( 16%)
2      11 ( 44%)
3       8 ( 32%)
```

Figure 5: A sample clustering result generated by WEKA

Table 5: Clustering results at different values of *k*

Value of k (Number of Clusters)	Clusters and their features
2	Cluster 0 = 3 (Resolution, Flash, Clarity) Cluster 1 = 22 (Camera, Picture, Lens, Zoom, Photos, LCD, Size, Color, Display, View-Screen, Optics, Button-design, Battery, Price, Video, Weight, Processing-time, Software, Functions, Share-software, Money, Wide – angle)

3	Cluster 0 = 2 (Resolution, Clarity) Cluster 1 = 12 (Camera, Picture, Lens, Zoom, Photos, LCD, Size, Color, Display, View-Screen, Optics, Button-design) Cluster 2 = 11 (Battery, Price, Video, Weight, Flash, Processing-time, Software, Functions, Share-software, Money, Wide –angle)
4	Cluster 0 = 2 (Resolution, Clarity) Cluster 1 = 4 (Size, Display, View-screen, Button-design) Cluster 2 = 11 (Battery, Price, Video, Weight, Flash, Processing-time, Software, Functions, Share-software, Money, Wide–angle) Cluster 3 = 8 (Camera, Picture, Lens, Zoom, Photos, LCD, Color, View-Screen)

On the basis of the clusters formed by the K-means algorithm, the results of the experiment are interpreted in the following manner. An interpretation of the clustering results obtained at k = 2 can be summarized as follows:

1. Cluster-0 has features whose importance is low as few reviewers have commented directly on these features. However, it is very evident in this grouping is that all the features are related to the camera operations.
2. Cluster-1 has all the features on which the reviewers have expressed positive opinions as well as negative opinions and these features are important from both users and manufacturers point of views.

Thus, it can be deduced that clustering at k = 2, does not give encouraging result as the clustering is done mostly on the frequency of the features.
Similarly, the interpretation of the clustering results at k = 3 can be summarized as follows:

1. Cluster-0 has resolution and clarity, whose frequency of occurrence is low, but with positive opinions. One notable point is that both of the features mapped to cluster-0 are interdependent as high resolution indicated higher clarity and vice-versa.
2. Cluster-1 contains almost all features that occur in most of the documents and the opinion expressed on this feature group is quite high and positive. The features in this cluster can be termed as *most important features* for any product. The features which is worth mentioning here are "LCD" with "View-screen", and "Picture" with "Photos" that are synonym of one another and they are rightly mapped to one cluster.
3. Cluster-2 features are the one whose overall *satisfaction level of the reviewers is low* as very few reviews used highly positive sentiment words like "excellent", "amazing", and "awesome". These features can be considered as critical features that could be improved by the manufacturer for better customers' satisfaction. The highlight of this group is the features like "Price" and "Money", "Software" and "Share-software", which rightly mapped to the same group having the same semantic meaning.

Finally, the interpretation of the clustering results at k = 4 can be summarized as follows:

1. Cluster-0 remains intact as in the previous case since no new features moves in or out from this cluster.
2. Cluster-1 has now four features that have carved their place from cluster-1 obtained at k = 3, as these features are the one whose importance with other group members like "picture", "lens", "zoom", "LCD", etc. is low, because not many reviewers have commented frequently on these features. However, most of the comments on these features are positive and so they can be described as *features having positive opinions but with low frequency.*
3. Cluster-2 is same as the previous case and so all these features that mapped to this group are features on which the users' satisfaction level is not high and can be termed as *features with low negative satisfaction level.* It also includes some features that have not been frequently commented resulting low frequency in the dataset.
4. Cluster-3 contains the features that are same as cluster-1 obtained at k = 3, and it can be termed as features that are mostly liked by the reviewers and the comments expressed by them are fairly high. These features are the *most important features* of the product and are liked by the reviewers.

It can be observed from the experimental results of clustering mentioned above that it is an interesting approach to identify feature patters in which clusters are formed by not only taking features as a discrete quantity for input, but taking their related opinions as an additional parameter for performing the clustering and identifying similar feature group addressing a particular aspect of end-users.

5. CONCLUSION

In this paper, a multi-attributed feature clustering mechanism has been proposed that exploits expressed opinions over features to classify them into clusters based on the concerns (positive or negative) of the end users. The proposed clustering mechanism can be used to group features into different coherent clusters that provide an insight on the behavior of the features expressed in users' reviews. The advantage of such a method is that the user can find features which are dependent on one another and lies in a group. This clustering mechanism can also help manufacturers who want to know the features on which the user satisfaction is high or low so that they may concentrate on them without going through pile of review documents.

6. REFERENCES

[1] Liu, B., Hsu, W., Ma, Y. 1999. "Prunning and Summarizing the Discovered Associations". In Proceeding of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99), pp. 125-134.

[2] Liu, B. 2007. "Web Data Mining-Exploring Hyperlinks, Contents, and Usage Data", Springer Series on Data-Centric Systems and Applications.

[3] Hu, M., Liu, B. 2006. "Opinion Features Extraction using Class Sequential Rules". In Proceedings of the Spring Symposia on Computational Approaches to Analyzing Weblogs.

[4] Liu, B., Hu, M., Cheng, J. 2005. "Opinion Observer: Analyzing and Comparing Opinions on the Web". In Proceedings of the 14th International World Wide Web Conference (WWW 05), pp. – 342-351.

[5] Ding, X., Liu, B., Philip. S.Y. 2008. "A Holistic Lexicon-Based Approach to Opinion Mining", In proceedings of the first ACM International Conference on Web search and Data Mining (WSDM'08), California, USA, pp. 231-240.

[6] Bodendorf, F., Kaiser, C. 2010. "Mining Customer Opinions on the Internet- A case study in the Automotive Industry". In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, pp. 24-27.

[7] Chaoji, V., Hoonlor, A., Szymanski. B. K. 2008. "Recursive Data Mining for Role Identification", In Proceedings of IEEE/ACM Fifth International Conference on Soft Computing as Transdisciplinary Science and Technology, pp.218-225.

[8] Balahur, A., Montoyo, A. 2008. "A Feature Dependent Method for Opinion Mining and Classification", In Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering, pp. 1-7.

[9] Lafferty, J., McCallum, A., Pereira, F. 2001. "Conditional Random Fields: Probabilistics Models for Segmentating and Labelling of Sequence Data". In Proceedings of the International Conference on Machine Learning (ICML '01), pp. 282-289.

[10] Freitag, D., McCallum, A. 2000. "Information Extraction with HMM Structures Learned by Stochastic Optimization". In Proceedings of National Conference on Artificial Intelligence (AAAI'00).

[11] Pang, B., Lee, L. 2004. "A Sentiment Education: Sentiment analysis using subjectivity summarization based on minimum cuts". In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 271-278.

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

[13] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009; The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[14] Kalashnikov, D. V., Chen, Z., Mehrotra, S., and Nuray-Turan, R. 2008."Web People Search via Connection Analysis". IEEE Transactions on Knowledge and Data Engineering, 20(11):1550-1565.

[15] Borgatti, S. P., Everett, M. G., and Freeman, L. C. 2002. UCINET 6 for Windows: Software for Social Network Analysis. Harvard: Analytic Technologies.

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