


# Opinion mining on large scale data using sentiment analysis and k-means clustering

Sumbal Riaz<sup>1</sup> · Mehvish Fatima<sup>1</sup> · M. Kamran<sup>1</sup>  · M. Wasif Nisar<sup>1</sup>

Received: 28 March 2017 / Revised: 13 July 2017 / Accepted: 25 July 2017 / Published online: 8 August 2017  
© Springer Science+Business Media, LLC 2017

**Abstract** With the rapid growth of web technology and easy access of internet, online shopping has been increased. Now people express their opinions and share their experiences that greatly influence new buyers for purchasing products, thereby generating large data sets. This large data is very helpful for analyzing customer preference, needs and its behavior toward a product. Companies face the challenge of analyzing this sheer amount of data to extract customer opinion. To address this challenge, in this paper, we performed sentiment analysis on the customer review *real-world data* at phrase level to find out customer preference by analyzing subjective expressions. Then we calculated the strength of sentiment word to find out the intensity of each expression and applied clustering for placing the words in various clusters based on their intensity. We also compared the results of our technique with star-ranking given on the same dataset and found the drastic change in our results. We also provide a visual representation of our results to provide a clear insight of customer preference and behavior to help decision makers for better decision making.

**Keywords** Heterogeneous data processing · Imbalanced learning · Intelligent computing

## 1 Introduction

Due to rapid advancement of web technology, the Internet has become a very important source of information for many people. This advancement has boosted up the e-commerce;

as a result, now people tend to take more interest in buying products online rather than moving around in town. In this context, customer reviews influence the purchasing decision of buyers. People usually share—by customer reviews—their opinion, attitude, feeling and emotion towards the product they buy online. These reviews can be very helpful for a new buyer for selection of product. This is also because “What other people think” has always been an important piece of information for most of us during the decision-making process. Moreover, this user-generated content is informative and valuable to business managers who are eager to learn how and in what aspects customers’ like or dislike their products and services. In the recent past, many researchers have evaluated product on star based ranking method, where customers have to give rating from one to five according to the product quality. However, sometimes those rating value does not match with the product review given in textual form. For example, some people show biasness in their opinions by rating the product 4/5 but in their text review they mention few negative opinions towards product too. Accordingly, for the detailed analysis of customer opinion, text mining has emerged to be one of the most suitable approaches.

According to the survey of [1] on 2000 American adults, 81% of internet users have done research about a product for at least one time. The impact of these review content influenced the purchase of about 80–87% companies like restaurants, hotels and their various services. Furthermore, customers prefer to pay more for the product that got 5 star rating and very positive review content. Consequently, this large number of reviews on e-commerce websites like Amazon eBay, social networking sites like Facebook twitter and blogging websites pose a great challenge for information and communication technology researchers to effectively mine the useful and relevant information.

✉ M. Kamran  
m.kamran.nuces@gmail.com

<sup>1</sup> Department of Computer Science, COMSATS Institute of Information Technology, Wah Cantt, Pakistan

Lately, various data mining and text mining techniques have been applied to effectively extract, transform, load and analyze this largely produced, structured and unstructured data consisting of reviews in the form of textual information. Broadly, this textual information is categorized into two main types: facts and opinions. Facts are the objective expressions about entities, events and their properties. Opinions are usually subjective expressions that describe people's sentiments, appraisals or feelings toward entities, events and their properties [2]. For the analysis of customer reviews, researchers and academicians are working rigorously on sentiment analysis for last one and half decade in many fields like healthcare [3], social media [4], politics [5], news and blogs [6]. These analysis techniques also include sentiment analysis (SA) that is a computational study of opinions, sentiments, emotions, and attitude expressed in texts towards an entity [7]. Sentiment analysis (also called opinion mining, review mining, appraisal extraction, and attitude analysis) is the task of detecting, extracting and classifying opinions, sentiments and attitudes concerning different topics as expressed in textual input.

Opinion mining and sentiment analysis research aims to know the opinions of users all over the Web through analysis of feedback (or review) text. In the recent past, many researchers have proposed their work on sentiment analysis done at the document level (for example, distinguishing positive from negative reviews). However, tasks such as multi-perspective question answering and summarization, opinion-oriented information extraction, and mining product reviews require sentence-level or even phrase-level sentiment analysis. Accordingly, recognizing semantic impact of words or phrases is a challenging task in itself and requires some human-intelligence along with sentiment analysis and mining. This is mostly because: (i) the data needs to be structured; (ii) the star-ranking made by customer might be biased; (iii) the text of the feedback should, therefore, should also be analyzed for clearer idea of the feedback; (iv) the text of the feedback requires natural language processing; and (v) the data, therefore, needs some human intelligence (this requires some manual scoring for assigning polarity values to the text) for handling the biasedness of the ranking through mining of feedback text.

To respond to such challenges, the major contributions of this paper are:

- We performed the sentiment analysis at phrase level to determine the sentiment of each phrase.
- We determined the polarity of each word by classifying each expression in positive, negative and neutral category.
- Then we find out the score value of each sentiment term by using keygraph keyword extraction technique.
- We also calculated sentiment strength of each term to determine its behavior with term frequency and polarity.

- At the final stage we performed clustering to cluster the data on the basis of its sentiment strength value.
- Experimental results were visualized by using interactive charts and diagrams.

Accordingly, the significance of the proposed work is that it will help the managers to know that how their products and services are perceived in market and also guide customers to find out the product that is most trendy now a day. Moreover, imbalanced nature of the customer reviews data for various products also present a unique challenge for the researchers for clustering and mining of the data. Consequently, mining of imbalanced datasets has also become a very motivating research area. In this work, we also considered this subject and gathered large scale *real-world data* of product reviews of six categories that was unstructured (and semi-structured) and imbalanced in nature. The proposed work will definitely help decisions makers to align their future plan in the targeted market by analyzing public mood regarding their product.

The rest of the paper is organized as follows. Section 2 presents the related works. Section 3 demonstrates the system framework for sentiment analysis on product reviews, keyword extraction and Sect. 4 describes the experiments, along with data collection and data analysis, followed by the experimental results and discussion. Finally, Sect. 5 concludes and portrays few directions for the future work.

## 2 Related work

Opinion mining and sentiment analysis research aims to mine the opinions of users available over the Web and their major applications include mining of product/service reviews [8], recommendation systems [9] business and government intelligence [10]. Sentiment classification aims to identify the sentiment (or polarity) of retrieved opinions. There are two categories of approaches for this task. First one is lexicon based [11, 12] that involves calculating the orientation for a document using the semantic orientation of words or phrases in the document [13]. The second approach is based on machine learning technique [14, 15] in which data is trained and deployed to a sentiment classifier, which can be built with several methodologies, such as support vector machine (SVM), maximum entropy, and naïve Bayes.

Supervised learning approaches require training data for sentiment classifier learning. Many micro blog and social networking websites obtain training data with their polarities (positive, negative, neutral) [16–18] or by taking consensus from sentiment detection websites [19]; moreover, supervised learning require retraining with the arrival of new data. Therefore, these approaches affect the emerging topics in this field with great limitations. Another limitation of supervised approaches is their domain dependence; for instance,

classifier trained on data from one domain (for instance, data relating to health reform) produce unsatisfactory performance when applied to data from a different domain [20] (for instance, data relating to products). The technique in [21], calculates sentiment polarity strength of a review by multiplying the strength of used adjectives and adverbs. The strength of an adjective is calculated using progressive relation rules of adjectives and link analysis (propagation algorithm). While the scheme in [22] uses a two-step based feature-level opinion mining and ranking algorithm which was deployed in a search engine AskUs. In [23], authors devised a supervised term weighting scheme based on importance of a term in a document (ITD) and importance of a term for expressing sentiment (ITS) with the help of 7 statistical feature selection methods including document frequency (DF), information gain (IG), mutual information (MI), odds ratio (OR), chi-square statistic (CHI), Weighted Log Likelihood Ratio (WLLR) and Weighed Frequency and Odds (WFO). In [24], the authors performed sentiment classification of tweets. They experimented with 11,875 manually annotated tweets. They employed five different combinations of features over unigram, senti-features, and tree kernel.

Lexicon-based methods try to overcome the aforementioned limitations by using the sentiment orientation of words and phrases in a given document to calculate its overall sentiment without the training data. Instead, they use lexicons of words weighted with their sentiment orientations and rely on sentiment lexicons, that is pre-built dictionaries of words with associated sentiment orientations such as SentiWordNet [21], MPQA subjectivity lexicon [22], or the LIWC lexicon [23]. These lexicons (although costly to obtain) once constructed, are applicable to a wide variety of domains. To reduce the cost of building these sentiment lexicons, some approaches apply bootstrapping techniques to add words to an initial subset or seeds. However, overall, these approaches have shown to work effectively on conventional text [2]. The technique in [25] applied 3 methods in tandem: Enhanced Emoticon Classifier (EEC), Improved Polarity Classifier (IPC), and SentiWordNet based classifier (SWNC) to classify tweets. The authors in [26] proposed a lexicon-enhanced method for improving the sentiment analysis of user generated reviews based on rule-based classification scheme. The proposed method used the emoticon classifier, modifier & negation classifier, SWN-based classifier (SWNC) and (DSC) domain specific classifier in a sequential way to classify the reviews as +ive, -ive or neutral accordingly; however, they did not account for biasedness in the ranking as depicted by comparing star-ranking and the ranking based on the product reviews. Similarly, [27] analyzed consumers' sentiments towards major worldwide brands such as IBM, Nokia, and DHL using qualitative and quantitative methodology to address issues like detection

of hidden patterns in consumers' sentiments towards global brands. Another lexicon enhancement technique discussed in [28] where a domain dependent polarity is created from the labeled reviews by adapting domain independent polarity lexicon. For this purpose, the polarity tendencies of the words by evaluating the mutual information with positive and negative reviews are extracted. This technique used the SWN and also improved the feature weighting schemes by using MMI. However, this scheme does not provide any mechanism to compare star-ranking and the ranking made by customer feedback.

Another approach in [29] analyzed sentiment of tweets using ontological engineering. After preprocessing of 667 tweets on mobile, they created domain ontology of tweets using a proposed algorithm. The authors in [30], exploited twitter as a platform to instruct an educational tool called Finch Robot to take a picture and reading temperature, where both instructions could be given in any order. Tweets were preprocessed and POS tagged using tweet tagger. They devised 15 rules on the basis of POS to extract different events, which were verified using ConceptNet. The diachronic phenomena of two different domains namely socio-political and sports on Google N-grams corpora has been studied in [31]. Analysis was performed 761 and 34 words from Socio-political domain and sport respectively. Epoch delimitation was performed on the basis of word distribution over certain periods of time. They also analyzed the opinion change phenomenon using the covariance between the frequencies of two or more terms over a certain period of time. Eight types of emotions were decided for 14,000 words using WNA based NRC Word-Emotion Association Lexicon (WNANRC) and Semeval 2007 Affective Text (SAT). The proposed methodology can be extended to predict future changes in society like the covariance between socialism and capitalism.

Further advancement in SentiWordNet can be seen in [32], which shows better performance over SentiWordNet. This technique creates a lexicon SentiFul which was created by exploiting WordNet-Affect database containing 2438 direct and indirect emotion-related entries based on nine different emotions. Table 1 shows the summary of above mentioned literature review.

### 3 Proposed approach

#### 3.1 Overview

The main architecture of our proposed approach is shown in Fig. 1. As shown, the process mainly consists of six parts: data acquisition, data transformation, sentiment analysis, construction of keyword, term frequency, and inverse document frequency calculation and clustering.

**Table 1** Comparison of lexicon based approaches

S. Nr	Author name	Year	objective	Technique	Level of analysis	Dataset	Accuracy achieved
1.	Lu et al. [21]	2010	Sentiment strength polarity	Propagation algorithm	Link based	Hotel Reviews	71.6%
2.	Eirinaki et al. [22]	2012	Opinion mining and ranking	Adjective count algorithm	Feature based	Vacuum and DVD players	97.1%
3.	Deng et al. [23]	2014	TF,IG,MI,OR	Term weighting scheme	Feature based	Multi domain data set	85.5
4.	Agarwal et al. [24]	2013	2 way and 3 way sentiment classification	SVM	–	Tweets	75.5
5.	Khan et al. [25]	2014	Sentiment classification	Enhanced emoticon classifier improved polarity classifier. SentiWordNet based classifier	–	Tweets	87.5
6.	MZ Asghar et al. [26]	2017	Lexicon enchantment	SWN-based classifier (SWNC) and (DSC)	–	Drug, car, hotel	80
7.	Mostafa [27]	2013	Consumer sentiment	Relative frequency word counts	Word level	tweets	–
8.	Asghar [28]	2015	Domain dependent lexicon creation	SWN and weight scheme	–	Car,drugs, hotel reviews	–
9.	Kontopoulos et al. [29]	2013	Ontological sentiment analysis	Opendover (web service)	Phrase level	Tweets	70%
10.	Bell et al. [30]	2014	Instruct educational tool	–	–	Tweets	–
11.	Popescu and Strapparava [31]	2014	Opinion change after time	Word-Emotion Association Lexicon	Word level	Socio-political and sports	–
12.	Neviarouskaya et al. [32]	2011	Enhanced SentiWordNet	Compounding of words, changing bases and affixes of opinion words	–	9 different emotions database	94.1



**Fig. 1** Architecture of the proposed scheme

In the data collection phase, we crawled various websites for collection of different product reviews. We also used online available dataset provided by Amazon. Then we combined all the data and converted it in one format. At the data transformation stage, we did all file conversion settings and handled missing attribute values. The most important step of our technique is to apply sentiment analysis on raw review dataset for finding useful insight by using the lexicon based approach. At the first stage tokenization, dictionary tagging and bag of word creation is performed. Then for the reduction of noisy data, various preprocessing techniques are applied. For the extraction of keywords from data, we applied key-graph keyword extraction technique. By using this technique we can extract those keywords that act as base for whole the document. At the final stage, we calculate sentiment strength and applied k-means clustering algorithm for better analysis.

### 3.2 Data acquisition

The use of online shopping is dynamic and the data about the products feedback is available in huge size, but scattered and unstructured and misclassified form. Different web based tools like web crawlers and spiders are available online for data collection. We build our customize crawler specially for gathering product reviews from different online shopping websites. We used it to create a copy of all the visited pages for later processing by a search engine that indexes the downloaded pages to provide fast searches. Crawling is copying what is on web pages and repeatedly checking the multitude of pages to see if they are changed and making a copy of any changes found. Once a spider has crawled web page, the copy that is made is returned to the search engine and stored in a data center. The main functionality of crawler is as follows.

1. It reads the URL of target shopping websites from the list of shopping websites source database (seed).
2. It extracts the online website URL addresses from source database and puts them in the target URLs that are used for further feed of the next step.
3. Download the required information: review title, reviewer author, reviewer ID, reviews rating, and review content, reviews date and product category of the all target URLs shopping websites.
4. Store reviews data into database of system.
5. Provide data for further data analysis.

We also used online available dataset<sup>1</sup> of product reviews collected from Amazon. We collected, in total, over 1.2 million of product reviews in which the products belong to 6 major categories: camera, laptop, mobile phone, tablets, TVs and video surveillance devices. These online reviews were posted by 0.9 million reviewers (customers) for 20,821 products. Each review includes the following information: (1) Review Title (2) Reviewer Author (3) Reviewer ID; (4) Reviews Rating; (5) Review Content; (6) Reviews Date; and (7) Product category. The data spans from January 2002 to December 2014.

### 3.3 Data transformation

In this step, the dataset is converted in to a format that can be used for the further analysis. As the data is in raw form, so we need to enhance its quality and perform some formatting. For this purpose, we used Opal CSV Converter<sup>2</sup> to convert all files into CSV format from JSON. Few records also contained missing values and those rows were eliminated manually. For a quick reference, Table 2 lists the symbols used in the paper.

The first phase is the input of different data files of product reviews. We concatenated six different categories of reviews into single file  $D_E$ . Then we renamed the column of  $D_E$  according to the respective data value. The string to document node converts the *review-content* column into documents and then the column filter is applied for selecting only documents for further analysis because it includes main review text. Algorithm 1 lists the steps of this phase.

#### Algorithm1. Data Transformation.

**Input:** Data in Excel reader  $D_E$

**Output:** Document  $D$

1. concatenate files of  $D_E$
2. rename columns of  $D_E$
3. convert string to document  $D$

### 3.4 Sentiment analysis

After performing the necessary data transformations, the next step is the sentiment analysis where we will find the semantic

<sup>1</sup> <http://times.cs.uiuc.edu/~wang296/Data/>.

<sup>2</sup> <https://opal-convert-json-to-csv-to-json.en.softonic.com/>.



**Table 2** Notations

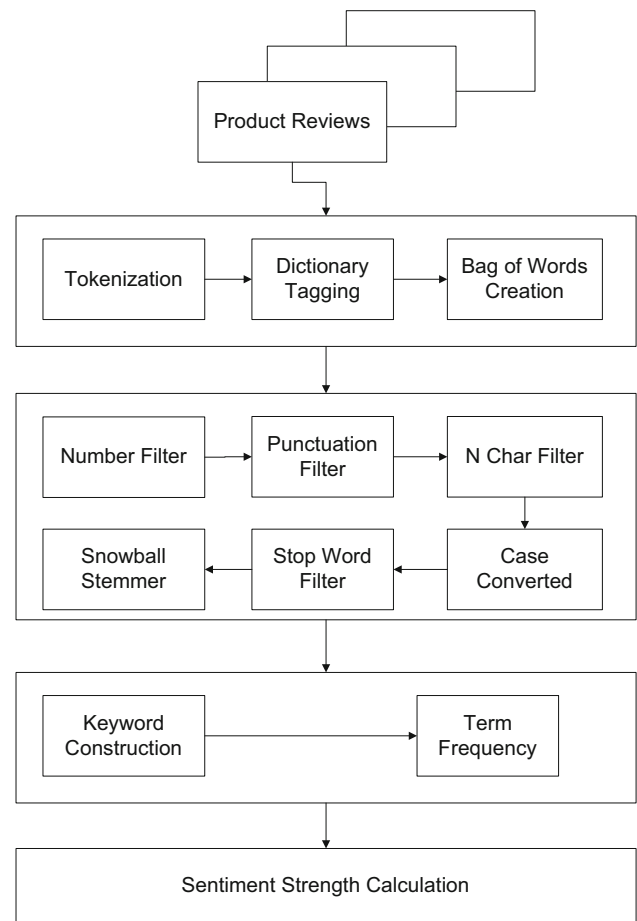
$D_E$	Data in excel reader
$D$	Document
$T$	Term
$T_p$	Positive term
$T_N$	Negative term
$T'_p$	Preprocessed positive term
$T'_N$	Preprocessed negative term
$d_p$	Dictionary of positive words
$d_N$	Dictionary of negative words
$C$	Category
$HF$	High frequency
$S$	Score
$S$	Sentence
$A$	Association
Key ( $w$ )	Key value
HFT	High frequency term
HKT	High key term
Col str	Column strength

orientation of review. This phase consists of several sub-phases as shown in Fig. 2 and explained below.

#### 3.4.1 Dictionary tagging and bag of word creation

In this step, by using Open NLP English word tokenizer,<sup>3</sup> each document is converted into terms.

After that, dictionary tagger is used where Multi Perspective Question Answering (MPQA) opinion corpus<sup>4</sup> is used for annotating subjective expressions.<sup>5</sup> We used MPQA cor-

**Fig. 2** Tasks performed in sentiment analysis

pus of both positive  $d_p$  and negative word  $d_N$  and tagged each word through dictionary tagger. Once the terms are tagged, a bag of word is formed for further preprocessing as shown in Algorithm 2.

#### Algorithm 2. Dictionary Tagging and bag of word creation

**Input:** Document  $D$ , Dictionary of positive words  $d_p$ , Dictionary of negative words  $d_N$ ,

**Output:** Bag of words

1. assign tag type=SENTIMENT to  $d_p$  and  $d_N$
2. assign tag value=POSITIVE to  $d_p$
3. assign tag value=NEGATIVE to  $d_N$
4. convert  $D$  into  $T$
5. compare  $T$  with  $d_p$  and  $d_N$
6.  $T_p \leftarrow T \cap d_p$
7.  $T_N \leftarrow T \cap d_N$

<sup>3</sup> <http://opennlp.sourceforge.net/models-1.5/>.

<sup>4</sup> [http://mpqa.cs.pitt.edu/corpora/mpqa\\_corpus/](http://mpqa.cs.pitt.edu/corpora/mpqa_corpus/).

<sup>5</sup> A subjective expression is any word or phrase used to express an opinion, emotion, evaluation, stance, and speculation.

#### 3.4.2 Pre-processing

In this phase, natural language preprocessing technique [33] is applied to extract only the tagged terms from a bag of words for further sentiment analysis.

- Number Filter

First number filter is applied to filter all terms consisting of numbers.

- Punctuation Erasure

Punctuation is used to create sense, clarity, and stress in sentences. But in our work, it does not provide any useful information; therefore, punctuations characters like period, comma, exclamation mark, question mark, colon semicolon, quotation, apostrophe, hyphen, dash, parentheses, and brackets are erased from each term by using punctuation erasure filter.

- N Char Filter

The N Char filter is applied to filter all the terms consisting of words with less than N characters. We set this value to “3” so all the terms with less than 3 characters (like is, a, so, ok) are filtered out.

- Case converter

The case converter changes the text into either the lowercase or the uppercase. We set this to “lowercase” option so all the terms are converted into lowercase.

- Stop word filter

Stop words are removed so that each line contains only one stop word.

- Snowball stemmer

Snowball stemmer [25] is applied to stem terms by the stemmer algorithms, this step reduces words to their stem or root form as stemming simplifies the sentiment analysis process. The same word can be used in a different flavor for grammatical reasons such as contain, contained, containing.

Algorithm 3 lists the steps involved in the data preprocessing stage.

**Algorithm3. Preprocessing.**

**Input:** bag of word containing T terms

**Output:** preprocessed positive terms  $T_p$ , preprocessed negative terms  $T_N$

1. Filter numbers from T
2. Filter punctuation character of T
3. Filter T contain character  $< N$
4. Convert T in small case
5. Remove stop word with T
6. Stem T

### 3.4.3 Assigning values to sentiment terms according to polarity

At this stage, we use “tag to string” that converts the term tag values of the SENTIMENT tag types to strings.

A column is appended in output table as SENTIMENT tag type, containing the strings representation of the corresponding tag value (that is positive sentiment or negative sentiment). The remaining terms that are not tagged by any positive or negative value, those terms are considered as neutral.

As some opinion words are not suitable to be categorized as “positive”, “negative”, or “neutral” because of their intensity towards the subject. So we assigned four different score values to sentiment terms by adopting the technique of [26] that is using manually crafted scheme for scoring sentiments. By using this technique, we assign the polarity score to the opinion words according to their intensity. For this purpose, we used rule engine where we defined different rules. Rules consist of a condition part (antecedent), which must evaluate to true or false, and an outcome (consequent, after the  $\Rightarrow$  symbol) which is put into the new column if the rule matches. We defined the rule that in “terms” column if most negative words like “not, never, nobody, neither” are present then set their polarity value “-2”, similarly in “terms” column if most positive words like “awesome, pretty, amazing” are present then set their polarity value “+2”. These terms are stored in separate column named as “new\_polarity” while remaining terms that have been already tagged with tag value “positive sentiment” and “negative sentiment” are given the polarity value as +1 and -1 and their values are stored in column names as “polarity”. Finally, we apply a condition for those terms that are not in “most positive, negative” category nor in “positive, negative” category and set the polarity of those terms as “0” and consider them as “neutral”. For example, consider a reviewer’s feedback as shown below:

“This tablet is amazing for running apps at home or using it as an e-reader. I was a little skeptical; considering the price; but it is a solid tablet. I originally bought it with the intent of rooting it; but even on the default Android OS; I am able to run everything i need from the Google play store. Obviously gps; led; and camera functions never work enough; but of course it was never designed for that. The kindle app works good which makes me wonder why anyone would get a different Android-based e-reader”.

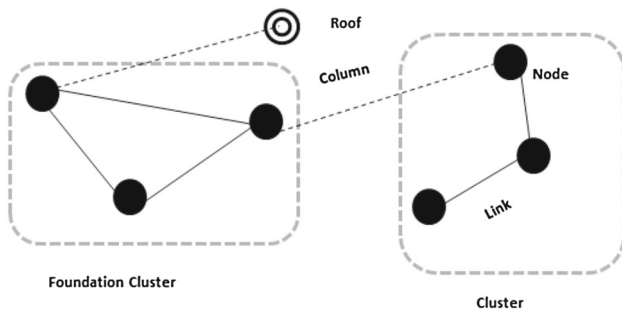
In this example, each word has been tagged with their polarity as found in the sentiment dictionary. Here the word *Amazing*, *Good* is tagged with positive polarity and *Skeptical* is tagged with negative polarity while *obviously* is tagged with neutral polarity and *never* is tagged with most negative polarity as shown below in Table 3.

### 3.5 Construction of keywords

Key graph keyword extraction is a technique for analyzing documents and extraction of relevant keywords using the graph-based approach described in [34].

**Table 3** Words with their polarity (an example)

Sentiment words	Word polarity	Polarity	New polarity
Amazing	Most positive	–	+2
Good	Positive	+1	–
Able	Positive	+1	–
Skeptical	Negative	–1	–
Obviously	Neutral	0	–
Never	Most negative	–	–2

**Fig. 3** Foundation extraction

This phase is composed of three major stages as discussed below.

### 1. Extracting Foundation

A graph  $G$  is formed on the basis of preprocessed document  $D$  with terms  $T$ . Each term in the document acts as a *node* and the *link* among the nodes is co-occurrence of the term in the same sentence. The graph is either singly connected or fully connected. A *singly connected graph* is one that connects graph with (number of nodes in  $G$ )-1 edges. In *fully connected graph* every node is connected to other node with unique link forming a foundation cluster. The terms in document are linked with the foundation cluster through column. The terms at the cross of strong columns are regarded as roofs.

Figure 3 shows the above relation diagrammatically.

The Link among the nodes is the association which is based on the co-occurrence of the term in the same sentence. We denoted the association of terms,  $t_i$  and  $t_j$  in  $D$ , as

$$assoc(t_i, t_j) = \sum_{s \in D} \min(|t_i|s, |t_j|s) \quad (1)$$

where  $|t_i|s$  denotes the count of the term  $t$  in the sentence  $s$ . This formulation comes from the assumption that each appearance of  $t_i$  is related to its nearest appearing  $t_i$ , not all  $t_j$  in the sentence.

### 2. Extracting column

We use Eq. (1) for extracting column to form relationship among the terms in the document and the basic cluster extracted. Keywords that connect two cluster are represented with some values  $key(t)$  that depict how strongly the clusters are held together. This value is calculated by using the following equation

$$Key(t) = 1 - \prod_{g \in G} \left( 1 - \frac{based(t, g)}{neighbor(g)} \right) \quad (2)$$

where  $g$  is the sub-graph of  $G$ ,  $based(t, g)$  represents number of occurrence of terms  $t$  in document  $D$  on the basis of cluster, while  $neighbor(g)$  represents count of the term  $t$  in the sentence including the terms in  $g$ .

### 3. Extracting roof

After assigning  $key(t)$  value to each term, the terms are sorted according to their association with cluster. Top  $N$  high key terms are selected and added in the graph. The foundation cluster contains low key terms (high frequency terms) but we do not exclude them as they are also important for summarizing document  $D$ . The strength of touching column make the high frequency term as well. Following equation measures the strength of the column

$$column(t_i, t_j) = \sum_{s \in D} \min(|t_i|s, |t_j|s) \quad (3)$$

The columns touching  $t_i$  are sorted by  $column(t_i, t_j)$  for each high key term  $t_i$  and the columns with the highest column values connecting term  $t_i$  to two or more clusters are selected to create new links in  $G$ . We show such links representing columns by dotted line in Fig. 4.

Finally, sum of touching column is calculated and nodes having sums higher than a certain threshold are extracted as the keywords for document  $D$ . For a testing purpose, we selected 12 top terms in our technique. This step helped us for extracting those keywords that act as a base for summarizing whole review. These selected keywords are used in the next phase for TF and IDF calculation.

An example of a customer review is mapped on this technique and presented in the form of a graph: “Good so far; still learning all its capabilities. Can’t access the web away from home or office; always requires a password for wireless access”.

For convenience, algorithm 4 lists all the steps involved in the keyword construction phase.



**Algorithm 4. Key Graph Keyword score calculation****Input:** document D, Term T**Output:** score value

1. Extracting foundations
2. Sort T with HF value>N
3. Link formation of T by calculation *assoc* a using Equation (1)
4. Extracting columns
5. Assign Key(w) to each term T in document D for tightness among cluster g
6. Compute Key(w) using Equation (2)
7. Extracting roof
8. Sortkey(w) and select N high key terms HKT
9. Strength calculation among HKT and HFT.
10. Compute Col str using Equation (3)
11. Keyword=Col str>threshold

**3.6 Term frequency and inverse term frequency for sentiment analysis**

The statistical method for finding the usage of term in document is Term Frequency—Inverse Document Frequency or simply TF-IDF. Its works by finding that how many times a term occurs within a specified document. It counts that term as positive value and adds a column containing the Tf value. This value is computed by dividing the absolute frequency of a term in a document with the number of all terms of in that document. We calculate the absolute term frequency using Eq. (4).

$$TF(t, d) = \begin{cases} 1 & \text{if } word = t \\ 0 & \text{else} \end{cases} \quad (4)$$

where  $t$  is term in document  $d$  and  $word$  represents the term  $t$ .

The inverse term frequency, a negative weight is assigned to the term relative to the number of documents that contain that term. We compute the probabilistic *idf* that is defined by using Eq. (5)

$$idf(t) = \log(f(D) - f(d, t)) / f(d, t) \quad (5)$$

where  $f(D)$  is the number of all documents and  $f(d, t)$  is the number of documents containing the term  $t$ .

The higher TF-IDF value indicates that a term is both important to the document as well as relatively uncommon across the documents. This is often interpreted to mean that the word is significant to the document and could be used to accurately summarize the document [35].

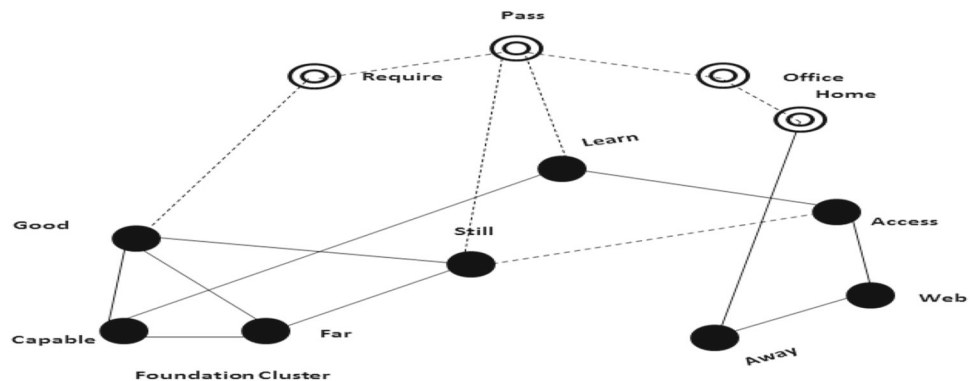
**3.7 Sentiment strength calculation**

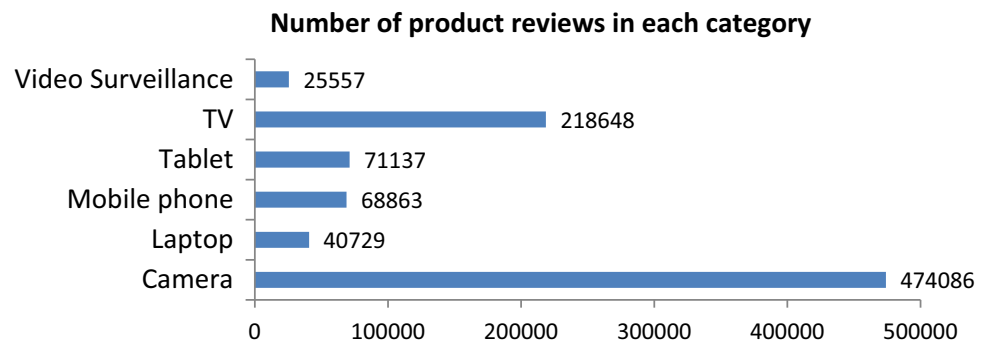
In this section, we provide the details about the method used to measure the strength value of the sentiment word. We compute the strength  $S$  of a term  $T$  by using its polarity value  $P$  new\_polarity  $P'$  and the term frequency  $TF$ . The formula for computing sentiment strength is:

$$S = P \times P' \times TF \quad (6)$$

where  $P$  value can be +1, 0, −1; while  $P'$  value can be +2, −2. The value of term frequency depends on the time of

**Fig. 4** Roof and column extraction



**Fig. 5** Data statistics

occurrence of the term in the document. The values of  $P$ ,  $P'$ , and  $TF$  are directly proportional to sentiment strength.

### 3.8 Clustering

In sentiment analysis, we used partition method to cluster the data on the basis of its sentiment strength value. In this approach a database  $D$  of sentiment strength value  $n$  is partitioned into  $k$  clusters so that sum of squared distances is minimized (where  $c_i$  is the centroid or medoid of cluster  $C_i$ ) using the relation given in Eq. (7).

$$E = \sum_{i=1}^k \sum_{p \in C_i} (d(p, c_i))^2 \quad (7)$$

In the partitioned method, we used k-means clustering algorithm where each cluster is represented by the center of cluster (centroid). Then we specified  $k$  value to form  $k$  number of clusters. Initially, the centroids of each cluster were selected randomly. After that data is divided in those  $k$  clusters. Then second step of Euclidean distance calculation is performed to determine the distance between each data point and centroid. The Euclidean distance of each data point should be minimum from its centroid. When all the data points have been included in cluster initial grouping was done. As the inclusion of data points altered the centroid, so we recalculated centroid of each cluster. This recalculation is continued until no more new centroid is formed and finally all the data points are grouped in their respective clusters.

## 4 Experimental study

This section provides the detail about the experiments conducted to validate the advantages of the proposed methodology. For sentiment analysis, we used the data in textual form that is available abundantly online in the form of reviews and the customer feedback.

### Data acquisition

We used online dataset of six products from Amazon and also collected more data by using crawler on shopping web-

**Table 4** Attributes of the data

Attributes	Data type	Description
Review title	String	Review in summarize form
Author	String	Name of reviewer
2	String	ID of reviewer
Stars	String	Product rating
Review content	String	Detail review text
Date	String	Review date
Category	String	Different product categories

sites like eBay and Alibaba. The dataset was collected on large scale and collected from January 2002 to December 2014. About 1.2 million reviews were grouped in six following categories: camera, mobile phone, laptop, tablet, TV, and video surveillance devices. In each domain category, we have subcategories of each product on the basis of its model and technology, but in this experiment, for brevity, we accumulated all in one major category. These online reviews were posted by 0.9 million reviewers (customers) towards 20,821 products under these six categories. Figure 5 shows the statistics of this data.

### Data transformation

In this step, we transformed the raw data in a format such that it is ready to be analyzed. Our collected data was in JSON file format that we converted in CSV format and then organized the data in the format having following attributes given in Table 4.

### Sentiment analysis

#### Dictionary tagging and bag of word creation

First tokenization of review text was performed by using Open NLP word tokenizer so that each document is converted into terms. We used MPQA corpus of both the positive and the negative word and tagged each word through dictionary tagger. Once the terms were tagged with POSITIVE and NEGATIVE tag value a bag of word was formed for further preprocessing. In Table 5, an analysis of an example text of a customer review is shown where every word of review content has been tagged

**Table 5** Analysis of a custom review (an example)

Review content	Terms with polarity
Very pleased with this purchase and it works beautifully	Very  Pleased (POSITIVE) with This Purchase And It Works Beautifully (POSITIVE)

### Data preprocessing

After tagging positive and negative terms, the bag of words is preprocessed to remove unnecessary information. For this purpose, various filters were used like number filter, punctuation erasure, N character filter, stop word filter, and snowball stemmer.

### Assigning values to sentiment terms according to polarity

At this level, all the noisy data is removed and only tagged terms are left. We used tag to string conversion and all the terms that were tagged through dictionary tagger under SENTIMENT tag type were converted into string representation of corresponding tag value (that is, most positive sentiment, most negative, positive, and negative sentiment). The remain-

ing terms that are not tagged by any tag are considered as neutral. The polarity values of each sentiment term are assigned according to the rule defined in Sect. 3.4.3.

As an example, Table 6 shows a customer review with only the preprocessed terms retained while remaining terms were filtered out.

### Key graph keyword extraction

In this step, keygraph keyword extraction technique was applied to extract a specific word on the basis of which whole document can be summarized. Equations (1), (2), and (3) were used and a score value of each term was obtained.

The output of this step is shown in Table 7 with different keywords with their score values.

### Term frequency and inverse document frequency calculation

In this step, absolute term frequency and inverse document frequency of each term according to its document was calculated by using Eqs. (4) and (5). By this method, we found out the number of occurrence of any word in a particular document.

### Sentiment strength calculation

We also calculated the strength of the sentiment term so that we can find the intensity of sentiment. We used Eq. (6) for strength measurement.

Table 8 shows strength of sentiment along TF-IDF values.

### Clustering

By using k-means clustering technique, we found the number of words in each cluster according to its sentiment

**Table 6** Analysis of a customer review with only preprocessed terms (an example)

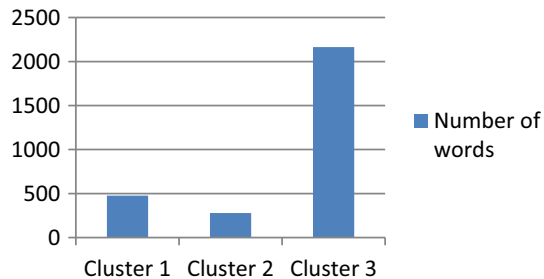
Review content	Preprocessed terms with sentiment	Polarity	New polarity
Love this tablet. Excellent purchase. I can do so many things with it. I can help my granddaughter learn her colors; numbers; and alphabet. Read books; play games; stay up on social media and news but unhappy with battery life	Love <b>MOSTPOSITIVE(SENTIMENT)</b>	–	+2
	Excel <b>MOST POSITIVE(SENTIMENT)</b>	–	+2
	Number <b>NEUTRAL</b>	0	–
	Alphabets <b>NEUTRAL</b>	0	–
	learn <b>POSITIVE(SENTIMENT)</b>	+1	–
	Play <b>POSITIVE(SENTIMENT)</b>	+1	–
	Unhappy <b>NEGATIVE(SENTIMENT)</b>	–1	–

**Table 7** Key graph score extraction and keywords score (an example)

Review content	Keyword	Score	Sentiment
This tablet has everything good	Good	8	Positive
I don't like i barely used it	Bare	21	Negative
I made the miserable mistake	Miserable	55	Negative

**Table 8** Sentiment strength calculation

Review content	Keyword	Sentiment	TF	IDF	Strength
Better than expected, much better	Better	Positive	4	1.87	4
Love it! Excellent choice	Love	Positive	1	1.204	1
Cheap unreliable garbage	Cheap	Negative	2	2.179	−2

**Fig. 6** Clustering of sentiment strength

strength value. The data points whose sentiment strength values are similar to each other were grouped in one cluster. As the selection of  $k$  value highly depends upon the performance of  $k$ -mean clustering algorithm. Therefore, we checked the output by selecting various values of  $k$  and chose the best one. First we set  $k = 2$  and analyzed the results. The data points within a cluster are very dissimilar to each other. As we know that, a good clustering approach is one where the data points within a cluster should be very close to each other. when we changed the  $k$  value and set that as 3 and when results was analyzed all data points within the clusters are similar to each other and different from other cluster's data point. So these results are good as compared to previous one. But when we set  $k = 4$ , our 4th cluster had no data point. So, we used  $k = 3$  and all the data points were grouped in their clusters with minimum Euclidean distance from their centroids. For handling the outliers, the word polarity and the star-ranking were used to handle the reviews which were very different as measured by star-ranking and sentiment analysis. For instance, if a customer assigned the star-ranking of 1 and his/her product review had words like “very good”; such reviews were counted as outliers. In our experimental study, we did not exclude such customer reviews because our approach identifies such interesting patterns for accounting customer biasedness. Figure 6 shows the result of clustering performed on a dataset of mobile category reviews.

The performance of the clustering algorithm was measured by using the performance metrics suggested in [36] by a researcher at Stanford. First of all, the purity of the cluster was computed by assigning the most frequent class to each cluster based on the classes (polarities) present in the cluster followed by measuring of accuracy by counting the number of correct assignments divided by total number of classes present in the cluster. Mathematically,

**Fig. 7** Tag cloud on television category**Table 9** Star-ranking

Star ranking	
1 star	Very bad
2 star	Bad
3 star	Neutral
4 star	Good
5 star	Very good

**Table 10** New ranking

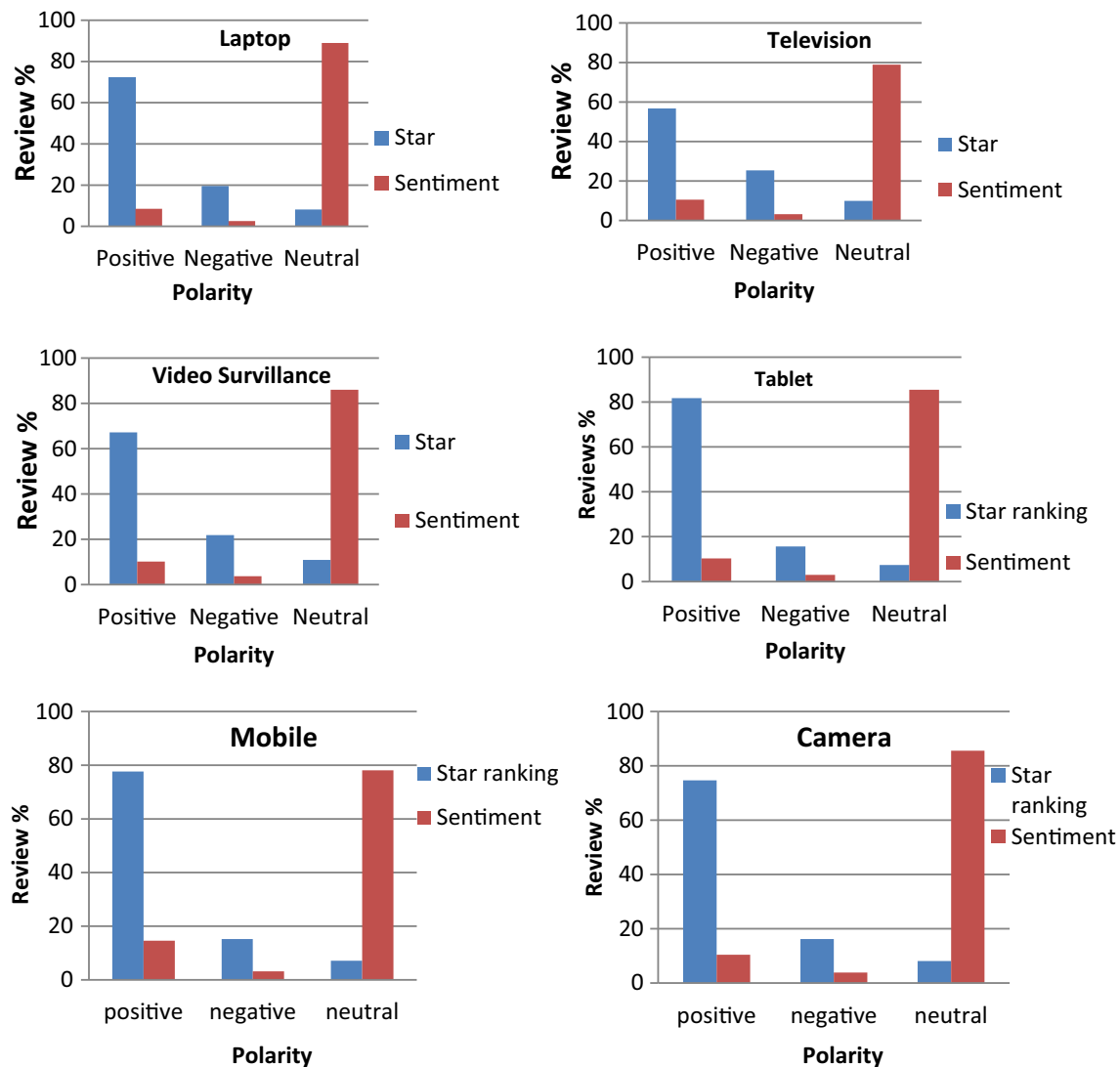
New star-ranking	
1 & 2	Negative
3	Neutral
4 & 5	Positive

$$purity(\Omega, C) = \frac{1}{N} \sum_{i=1}^k \max_j |\omega_k \cap c_j| \quad (8)$$

with,  $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$  representing the set of clusters and  $C = \{c_1, c_2, \dots, c_j\}$  showing number of classes

The value of this purity was between 1 and 0 with a value closer to 1 meaning the clustering was good and  $z$  value closer to 0 means the results were not good. We achieved the value of 0.97 for the same data as shown in Fig. 6. Moreover, the Rand Index ( $RI$ ) was calculated as follows:

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \quad (9)$$



**Fig. 8** Sentiment analysis

The obtained result for  $RI$  was 0.70 (rounded to two decimal places). Finally, the F-measure (involving precision and recall) was calculated as 0.49 (rounded to two decimal places). For the comparison purpose, we investigated the performance of our technique and the performance of a recent technique proposed in [32]. For this experiment, the same dataset of Fig. 6 was used. The  $RI$  (or accuracy) for the proposed technique and the technique of [32] was found to be 0.95 and 0.94 respectively. Although, the performance of both the techniques is almost same; however, the technique of [32] does not provide any mechanism to compare the star-ranking and the customer review-based ranking. Therefore, our claim about the novelty of our work regarding account of biasedness of the customer in product review as depicted by star-ranking and sentiment analysis is empirically evaluated using this experiment.

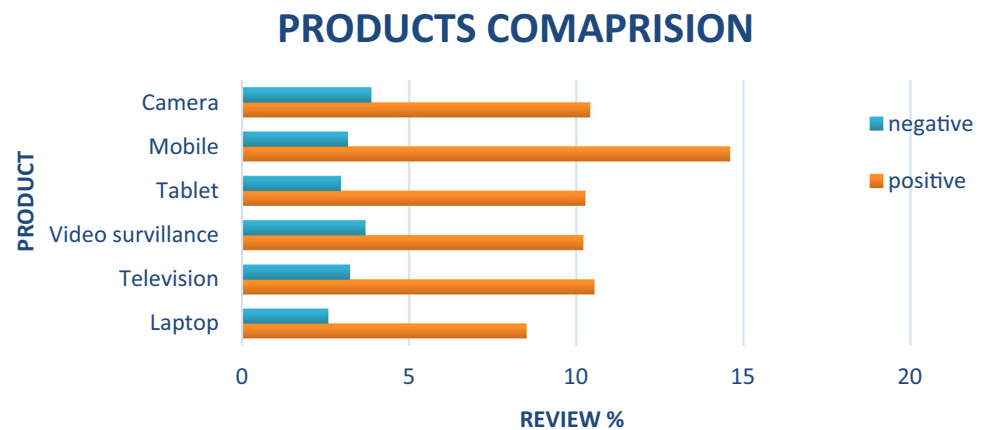
We also visualized these terms with the help of tag cloud. In tag cloud, the terms with green color are positive sentiment and the terms with red color (see Fig. 7) are negative sentiment. Also the size of the term indicates higher value of the term sentiment strength.

## Discussion on results

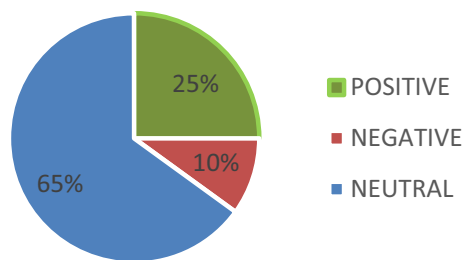
The final result of our technique with neutral, positive, and negative polarity on all dataset is discussed in this section. We compared the results of star-ranking and our results produced by sentiment analysis on different categories of product reviews with the one given by that customer on same review. For this purpose, we analyzed “star-ranking” values and transform them in order to compare. The star-ranking is based on 5 values as shown in Table 9. We combined them



**Fig. 9** Comparison among reviews of various products



### Sentiment Polarity



**Fig. 10** Sentiment polarity

and made three categories and assigned “+1” value to positive category, “−1” value to negative category and “0” to neutral category. The transformed star-ranking is shown in Table 10.

We also compared the results of star-ranking and sentiment analysis of all categories. Figure 8 shows the bar charts of 6 six categories. We found that the results from sentiment analysis are more neutral than polar. While we performed the sentiment analysis on the detail text content given by the customer, we examined the polarity of each word given in the review text.

From among the positive, negative, and neutral sentiment, we extracted only positive and negative sentiment percentage that clearly shows all products result in Fig. 9 because positive and negative sentiment polarity plays important role to find out the customer preference toward products. Our result depicts that *mobile* category has highest positive sentiment polarity among all the products and *laptop* category has lowest negative sentiment polarity. Thus, we concluded that mobile vendors enjoy high customer preference as compared to other products.

When we combined the result from all categories, we found the division of sentiment polarity as shown in Fig. 10. It is evident from Fig. 10 that the majority reviews given

in all data were “neutral” in nature according to our proposed sentiment analysis method.

### 5 Conclusion

In this paper, we have proposed a technique of text mining for analyzing customer review to find the customers’ opinion. We performed the sentiment analysis on the large scale dataset of product (6 categories) reviews given by various customers on the internet. In our approach, sentiment analysis was applied at phrase level rather than document level to calculate every term’s sentiment polarity. Then keygraph keyword extraction technique was used for extracting keywords from each document with high frequency terms. Also the term frequency and inverse document frequency of each word was calculated to find out the number of occurrence of every word in the document. So we found the list of most commonly used words. Then we calculated the intensity of sentiment polarity by measuring its strength. For summarization and understating purpose of analyzed data, we used k-means clustering algorithm and grouped the data on the basis of sentiment strength value. We also compared the results of our technique with star rating of same data and found better and neutral sentiment toward products. As the star based rating give sometimes biased results; by analyzing the text of reviewer through this approach the sentiment of each term can be obtained. When we compared the results of each category, we found laptop with high positive sentiment and camera with high negative sentiments. Overall our approach categorized majority reviews as neutral. Our work will help new customers if they are purchasing these products for the first time and it will also facilitate managers and decision makers of company to find out how customers’ like or dislike their products. In future, we will work on the attributes of products more precisely to know that how people express their opinion.

## References

- Smith, A., Anderson, M.: Online Shopping and E-Commerce. Pew Research Center, Washington, DC (2016)
- Liu, B.: Sentiment analysis and subjectivity. In: Indurkha, N., Damerau, F.J. (eds.) *Handbook of Natural Language Processing*, 2nd edn. CRC Press, Boca Raton (2010)
- Asghar, M.Z., Ahmad, S., Qasim, M., Zahra, S.R., Kundi, F.M.: SentiHealth: creating health-related sentiment lexicon using hybrid approach. *SpringerPlus* **5**, 1139 (2016)
- Hutto, C.J., Gilbert, E.: Vader: a parsimonious rule-based model for sentiment analysis of social media text. In: Eighth International AAAI Conference on Weblogs and Social Media (2014)
- Wang, H., Can, D., Kazemzadeh, A., Bar, F., Narayanan, S.: A system for real-time twitter sentiment analysis of 2012 US presidential election cycle. In: *Proceedings of the ACL 2012 System Demonstrations*, pp. 115–120 (2012)
- Nielsen, F.A.: A new ANEW: evaluation of a word list for sentiment analysis in microblogs (2011). [arXiv:1103.2903](https://arxiv.org/abs/1103.2903)
- Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: a survey. *Ain Shams Eng. J.* **5**, 1093–1113 (2014)
- Bai, X.: Predicting consumer sentiments from online text. *Decis. Support Syst.* **50**, 732–742 (2011)
- Cambria, E., Schuller, B., Xia, Y., Havasi, C.: New avenues in opinion mining and sentiment analysis. *IEEE Intell. Syst.* **28**, 15–21 (2013)
- Archak, N., Ghose, A., Ipeirotis, P.G.: Deriving the pricing power of product features by mining consumer reviews. *Manag. Sci.* **57**, 1485–1509 (2011)
- Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. *Comput. Linguist.* **37**, 267–307 (2011)
- Kang, H., Yoo, S.J., Han, D.: Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Syst. Appl.* **39**, 6000–6010 (2012)
- Wang, S., Li, D., Song, X., Wei, Y., Li, H.: A feature selection method based on improved Fisher's discriminant ratio for text sentiment classification. *Expert Syst. Appl.* **38**, 8696–8702 (2011)
- Ravi, K., Ravi, V.: A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowl. Syst.* **89**, 14–46 (2015)
- Ye, Q., Zhang, Z., Law, R.: Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Syst. Appl.* **36**, 6527–6535 (2009)
- Kouloumpis, E., Wilson, T., Moore, J.D.: Twitter sentiment analysis: the good the bad and the OMG!. *ICWSM* **11**, 164 (2011)
- Saif, H., He, Y., Alani, H.: Semantic sentiment analysis of twitter. *Semant. Web-ISWC* **2012**, 508–524 (2012)
- Barbosa, L., Feng, J.: Robust sentiment detection on twitter from biased and noisy data. In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 36–44 (2010)
- Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 347–354 (2005)
- Pennebaker, J.W., Mehl, M.R., Niederhoffer, K.G.: Psychological aspects of natural language use: our words, our selves. *Ann. Rev. Psychol.* **54**, 547–577 (2003)
- Lu, Y., Kong, X., Quan, X., Liu, W., Xu, Y.: Exploring the sentiment strength of user reviews. In: *International Conference on Web-Age Information Management*, pp. 471–482 (2010)
- Eirinaki, M., Pissal, S., Singh, J.: Feature-based opinion mining and ranking. *J. Comput. Syst. Sci.* **78**, 1175–1184 (2012)
- Deng, Z.-H., Luo, K.-H., Yu, H.-L.: A study of supervised term weighting scheme for sentiment analysis. *Expert Syst. Appl.* **41**, 3506–3513 (2014)
- Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: *Proceedings of the Workshop on Languages in Social Media*, pp. 30–38 (2011)
- Khan, F.H., Bashir, S., Qamar, U.: TOM: Twitter opinion mining framework using hybrid classification scheme. *Decis. Support Syst.* **57**, 245–257 (2014)
- Asghar, M.Z., Khan, A., Ahmad, S., Qasim, M., Khan, I.A.: Lexicon-enhanced sentiment analysis framework using rule-based classification scheme. *PLoS ONE* **12**, e0171649 (2017)
- Mostafa, M.M.: More than words: social networks' text mining for consumer brand sentiments. *Expert Syst. Appl.* **40**, 4241–4251 (2013)
- Asghar, M.Z., Khan, A., Ahmad, S., Khan, I.A., Kundi, F.M.: A unified framework for creating domain dependent polarity lexicons from user generated reviews. *PLoS ONE* **10**, e0140204 (2015)
- Kontopoulos, E., Berberidis, C., Dergiades, T., Bassiliades, N.: Ontology-based sentiment analysis of twitter posts. *Expert Syst. Appl.* **40**, 4065–4074 (2013)
- Bell, D., Koulouri, T., Lauria, S., Macredie, R.D., Sutton, J.: Microblogging as a mechanism for human-robot interaction. *Knowl. Syst.* **69**, 64–77 (2014)
- Popescu, O., Strapparava, C.: Time corpora: epochs, opinions and changes. *Knowl. Syst.* **69**, 3–13 (2014)
- Neviarouskaya, A., Prendinger, H., Ishizuka, M.: SentiFul: a lexicon for sentiment analysis. *IEEE Trans. Affect. Comput.* **2**, 22–36 (2011)
- Asghar, M.Z., Khan, A., Ahmad, A., Kundi, F.M.: Preprocessing in natural language processing. *Emerg. Issues Nat. Appl. Sci.* **10**, 152–161 (2013)
- Ohsawa, Y., Benson, N.E., Yachida, M.: KeyGraph: automatic indexing by co-occurrence graph based on building construction metaphor. In: *Proceedings. IEEE International Forum on Research and Technology Advances in Digital Libraries: ADL 98*, pp. 12–18 (1998)
- Lee, D., Jeong, O.-R., Lee, S.: Opinion mining of customer feedback data on the web. In: *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, pp. 230–235 (2008)
- <https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html>. Accessed 20 May 2017



**Sumbal Riaz** received the BS degree in software engineering from Fatima Jinnah Women University, Rawalpindi, Pakistan, in 2015. Currently, she is working towards her MS thesis from COMSATS Institute of Information Technology. Wah Campus, Wah Cantt, Pakistan. Her research interests include data mining, big data analysis.



**Mehvish Fatima** received the BS degree in Computer science from University of Wah, Wah Cantt, Pakistan, in 2012. Currently, she is working towards her MS thesis from COMSATS Institute of Information Technology, Wah Campus, Wah Cantt, Pakistan. Her research interests include data mining and business intelligence, big data analysis, software Engineering, Decision theory, Cognitive computing, edge computing and Web Applications.



**M. Wasif Nisar** received PhD degree in Engineering Computer Science and Technology from Institute of Software, GUCAS China in 2009. He received his BSc degree in 1998 and MSc degree in Computer Science in 2000 from University of Peshawar, Pakistan. His research interest includes software estimation, software process improvement, distributed systems, data mining, CMMI-based project management and algorithmic.



**M. Kamran** received the M.S. and Ph.D. degrees in computer science from the National University of Computer and Emerging Sciences, Islamabad, Pakistan, in 2008 and 2012, respectively. Currently, he is an Assistant Professor with the Department of Computer Science, COMSATS Institute of Information Technology, Wah Cantt, Pakistan. His research interests include machine learning, evolutionary computation techniques, data security, health informatics, big data analytics, and decision support systems.