ACSIJ Advances in Computer Science: an International Journal, Vol. 2, Issue 4, No.5 , September 2013
ISSN : 2322-5157
www.ACSIJ.org

A Novel Feature Cloud Visualization for Depiction of Product Features Extracted from Customer Reviews

Tanvir Ahmad

Department of Computer Engineering,
Jamia Millia Islamia, (A Central University),
New Delhi, INDIA.
tahmad2@jmi.ac.in

Abstract

There has been an exponential growth of web content on the World Wide Web and online users contributing to majority of the unstructured data which also contain a good amount of information on many different subjects that may range from products, news, programmes and services. Many a times other users read these reviews and try to find the meaning of the sentences expressed by the reviewers. Since the number and the length of the reviews are so large that most the times the user will read a few reviews and would like to take an informed decision on the subject that is being talked about. Many different methods have been adopted by websites like numerical rating, star rating, percentage rating etc. However, these methods fail to give information on the explicit features of the product and their overall weight when taking the product in totality. In this paper, a framework has been presented which first calculates the weight of the features depending on the user satisfaction or dissatisfaction expressed on individual features and further a feature cloud visualization has been proposed which uses two level of specificity where the first level lists the extracted features and the second level shows the opinions on those features. A font generation function has been applied which calculates the font size depending on the importance of the features vis-a-vis with the opinion expressed on them.

Keywords: *Opinion Mining, Natural Language Processing, Feature Cloud, Visualization.*

1. Introduction

The growth of the World Wide Web has been so tremendous during the last one decade that it has contributed to generation of large amount of online data which are mostly in the form of unstructured or semi-structured in nature. This user generated content has contributed to the problem of *information overload*, from which distillation of knowledge is a challenging task as it involves the intricacies of natural language processing augmented with the complexity of users writing incorrect English, wrong punctuation marks and using abbreviations. As a result, there is a need for converting the information embedded in the free flowing text into structured form generally termed as *database curation*, without which the knowledge cannot be assimilated in a meaningful manner. Once the data is converted into structured form all conventional data mining algorithms can be applied by tuning it in the problem domain. The problem of feature cloud visualization can be viewed as three separate problems which have to be solved one after another in order to generate the cloud. In the first place the problem is to identify the features and their opinions from the text and then to convert it in a structured form. As a part of the second step these features have to be analyzed and the weight of each feature have to calculated so that the features are ranked based on their importance as expressed by the users both for the features on whose negative and positive sentiments have been expressed separately. In the final and the third step a font size generation method should be used so that these features are converted in a two layer feature cloud which shows the importance of features along with their opinions. In order to achieve these tasks separate methodology are applied at each step.

In this paper, the feature clouds are generated by first identifying the features, opinion and modifier $<f,m,o>$ triplet where f stands for feature, o for opinion and the optional m which stands for modifier expressed on the opinion o. The list of extracted triplets is then stored and pattern weight calculations are performed which will rank the features into two lists: positive feature list and negative feature list based on the opinion expressed on them. The total cumulative weight of the feature is calculated as the sum of its cumulative weight in the positive and the negative list. Thereafter, the feature cloud is generated where the center node is the product on which the features have been generated and the nodes are connected to it around the center item in such a way that all the features are directly connected to the center item and all opinion expressed on the features form a ring around the features. The font size of the features and their related opinion are proportional to the weight of the features.

In order to check the efficacy of the proposed framework the algorithm has been tested on two different dataset domains, Digital camera and Hotels, and the accuracy of the features extracted from them are evaluated and have been found to have comparable result irrespective of the domain of the documents. However some domains may require tuning of the rules in order to increase the efficiency of the system.

2. Related Works

Good visualization techniques are always a welcome addition to improve the readability of the whole process of information extraction so that the end user has the various options of looking things from different angle. The problem of searching for relevant information in a large collection of data is a very common activity, and this problem was noticed way back by Maron et al. [1]. There has been a lot of effort to retrieve precise and informative data which can be presented in a concise form for visualization. One way to achieve this target is through representation of textual data as a tag-cloud which have been studied by Koutrika et al. [2], Kuo et al. [3], and Venetis et al. [4]. A tag-cloud is a collection of main terms that are mined from voluminous texts and are presented in a pictorial way as a cloud of terms emphasizing them in order of their relevance. This has an advantage that the readers can very easily comprehend the relevance of the text data very quickly and can decide whether it is interesting to them or not. The two major factors including font size and placement emphasize the tags. Wordle [3] uses a random placement scheme in which the font size is determined by using the frequency of tags and words. Most of the techniques use the font size of a feature by calculating its frequency in the respective documents. The difference however lies in their placements. Some techniques place tags horizontally in the same order as they appear in the actual text documents.

Work on opinion mining started initially with many researchers with identification of opinion words which identifies adjectives such as good, excellent, bad and then applying these words on different domains in order to get the semantic orientation. A sizeable number of papers mentioning *sentiment analysis* focus on the specific application of classifying customer reviews as to their polarity – *positive* or *negative* [5,6].

To obtain detailed aspects, feature-based opinion mining is proposed in literature [7,8]. In [6], a supervised pattern mining method is proposed while in [7,8], an unsupervised method has been discussed. A lexicon-based approach has been shown to perform quite well in [7,9]. The lexicon-based approach basically uses opinion words and phrases in a sentence to determine the orientation of an opinion on a feature. The classification approach of customer reviews based on existing domain-specific corpus by applying a lexicon based sentiment analysis has been discussed in [10]. Rule based method has been used by us in [11] which extracts the features, opinions and the modifiers from the documents and has formed the basis of this paper for the first part of the problem. Another paper [12] incorporated one more rule in order to increase the number of features that can be extracted where the previous rules are not sufficient to extract them based on the semantic of the sentences.

3. Proposed Feature Cloud Visualization Framework

In this section, the complete design of the feature cloud visualization framework has been proposed in Figure 1 which contains the various sub-sections. The proposed framework consists of the following key functionalities sub-modules – *Document Pre-processing, Subjective/ Objective Analyzer, Document Parser, Feature and Opinion Learner and Pattern Weight Calculation and Pattern Cloud Visualization.* Further details of the functionalities are presented in the following sub-sections.

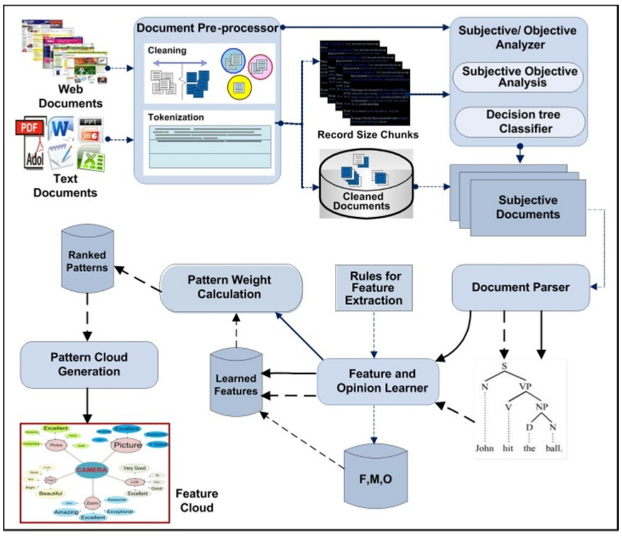

Fig. 1: Proposed Framework for Feature Cloud Visualization

3.1. Document Pre-Processing

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 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. Analysis had shown that subjective sentences are most likely to contain opinions.

3.2. Subjective/Objective Analyzer

According to Pang and Lee [9] 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 [8] 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. We have used a corpus of subjective and objective sentences used in [15] 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[1] [13] is trained to classify the unseen review sentences into subjective and objective classes. Weka (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. Weka is free software available under the GNU General Public License. Weka supports several standard data mining tasks, data pre-processing, clustering, classification, regression, visualization, and feature selection.

3.3. Document Parser

This module is responsible from extracting product features and opinions expressed on those features. It also extracts the optional modifier which may be a part of the review. This modifier is used to intensify the opinion expressed on the reviews and is an important part of the feature calculation process. However, as a first part of this exercise the subjective sentences extracted from the subjective/objective analyzer is 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 extract different types of hidden opinions inside the review documents. Each sentence of the dataset is converted into dependency tree which will be used to extract the $<f,m,o>$ from it.

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:

"The battery life of Nokia is very good."

In the above sentence, "battery life" is a noun phrase and appears as one of the features of Nokia phone 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 was implemented by us as a rule-based system [11, 12] 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 $POS(w_j) \neq$ NN* or $POS(w_j) \neq$ 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.

[1] http://www.cs.waikato.ac.nz/~ml/weka/

3.5 Pattern Weight Calculation

Pattern (in our case it is *feature*) weight calculation is done to grade the features of a particular product that reflects their degree of importance or relevance with respect to the given text corpus. This is mainly useful to facilitate end-users to easily comprehend the relevant features extracted from text documents. The feature weight calculation process can be informally summarized in the step as follows:

1. Compile the list of all feature words extracted from the corpus in the form of an information triplet containing feature, modifier, and opinion.
2. Calculate frequency count of each feature with respect to the corpus.
3. Determine the polarity of each opinion using Senti-WordNet and determine its class as negative, positive or neutral, along with the polarity score value.
4. Find the total weight for each feature by taking into account all positive and negative opinions expressed over it.
5. Arrange positive and negative features separately in descending order of the weights.

In order to perform the second and third steps mentioned above, it is required to calculate the frequency count of each feature and related modifiers along with the positive as well as negative sentiments. Though Senti-WordNet provides positive or negative score for a word, unfortunately it does not provide score for multi-word phrases, which generally occurs as modifiers are sometimes associated with opinion words. Therefore, fuzzy logic connectives such as AND, OR and NOT, and fuzzy quantifiers are used to calculate the overall weight of a feature, as shown in equations 1 to 5.

$$a\ OR\ b = \mu(a) \vee \mu(b) = \max(\mu(a), \mu(b)) \quad (1)$$

$$a\ AND\ b = \mu(a) \wedge \mu(b) = \min(\mu(a), \mu(b)) \quad (2)$$

$$NOT\ \ a = 1 - \mu(a) \quad (3)$$

$$VERY\ \ a = Q_{very}(\mu(a)) = [\mu(a)]^2 \quad (4)$$

$$FAIRLY\ \ a = Q_{fairly}(\mu(a)) = [\mu(a)]^{1/2} \quad (5)$$

Let f_i be the set of features extracted from a document corpus and T_i be the set of triplets that has been extracted for each f_i as expressed in equation 6.

$$T_i = \{T_{i1}, T_{i2}, T_{i3}, \ldots\ldots.., T_{in}\} \quad (6)$$

In order to determine the weight of a feature, we use equation 7 and 8 to calculate the overall weight of the feature with respect to all extracted triplets across the dataset. In equation 8, SWNScore is the Senti-WordNet score of the opinion words associated with the feature under consideration.

$$weight(f_i) = \sum_{j=1}^{|T_i|} (weight\ (T_{ij})) \quad (7)$$

$$tWeight(f_i) = freq(T_{ij}) \times T_{ij}\,[m](max\{SWNScore(T_{ij}\,[O])\} \quad (8)$$

The orientation of each feature as it appears in each triplet is calculated by using equation 9, where, f^z is a fuzzy function quantifier. The total weight of each feature is consolidated using equation 10, where, n is the number of opinions expressed over f_i.

$$orientation(o_i) = f^z\,(max\,\{sw^+(o_i)\,, sw^-(o_i)\,, sw^0(o_i)\}) \quad (9)$$

$$w(f_i) = \sum_{i=1}^{n} orientation(o_i) \quad (10)$$

In case a triplet has no qualifier or the fuzzy function of the extracted qualifier is not known the weight of the feature is calculated using the Senti-WordNet value of the opinion component only.

Table 1: A partial list of information triplets <f,m,o>

Sentence No.	Feature	Modifier	Commented word
10	Focus	-	Ultra-close
10	Modes	-	User-definable
22	Camera	-	Great
23	Zoom	-	Optical
48	Picture	very	Good
48	Picture	-	Easy
83	Wide angle	quite	Easy
101	Garbage	-	Canon
181	Size	-	Small
181	Camera	really	Nice

A frequency count of all features is estimated from the data structure maintained in table 1, which finds the number of sentences in which the corresponding triplet appears. The features on which positive and negative opinion has been expressed are segregated depending on the score of the Senti-WordNet. A partial list of features, termed as positive based on the opinion expressed on them, is shown in table 2. The score of the features containing the modifiers *very* and *little* is calculated using the fuzzy quantifiers as described in equation 1 to 6. Some of the features have been termed as negative as it contains opinions that are negative but they are very small in the domain of the digital camera dataset but their presence have to be taken into consideration while calculating the cumulative weight of that feature. This is obvious as reviewers often end up in giving different opinions on the same features based on their satisfaction/ dissatisfaction.

Table 2: A partial list of features along with negative opinions and their polarity score values

Feature (f_i)	Opinion Word (O_i)	Modifier (m_i)	SWNS	Fuzzified score	No.of sentences	W(f_i)	Cumulative weight of W(f_i)
LCD	Excellent	-	1.0	1.0	317	317.00	376.86
LCD	Good	Very	0.75	0.56	56	31.36	
LCD	Good	-	0.75	0.75	38	28.50	
Zoom	Excellent	-	1.0	1.0	65	65.00	108.13
Zoom	Exceptional	-	0.25	0.25	8	2.00	
Zoom	Awesome	-	0.875	0.875	47	41.13	
Picture	Excellent	-	1.0	1.0	457	457.00	578.50
Picture	Good	-	0.75	0.75	162	121.50	
Price	High	-	-0.25	-0.765	37	-28.30	-35.30
Price	High	Little	-0.25	-0.50	14	-7.0	
Battery Life	Short	-	-0.625	-0.625	32	-20.0	-20.0
Picture	Blurry	-	-0.75	-0.75	10	-7.5	-7.5
LCD	Small	-	-0.375	-0.375	8	-3.0	-3.0
Zoom	Small	-	-0.375	-0.375	23	-8.63	-13.38
Zoom	Average	-	-0.25	-0.25	19	-4.75	

On analysis, we found that some of the features qualify both as a positive feature and as a negative feature, which is truly justified as different users have expressed different opinions (positive or negative) on each feature thus bringing it in both categories. While calculating the overall weight of a feature, we calculate the total weight of the feature in the positive feature list, and subtract the negative weight if it also appears as a negative feature. If the total cumulative weight of the feature comes to be positive then it is termed as positive, otherwise a negative weight brings it to negative feature list. If a feature has a zero weight then it is termed as neutral feature and it is removed from overall calculation and this feature does not find a place in any of the list. However, such features need to be preserved as they provide the information that these features are neutral in nature and the end-users are not bothered while looking at the feature orientation, but it is a genuine feature of the product.

The total cumulative weight of a feature is calculated as the sum of its cumulative weight in the positive and negative list. After calculation of the total weight for all features they are ranked in their respective categories – *positive features* and *negative features*. This step requires sorting of the features on descending order of their weights. Tables 3 and 4 provide a partial ranked-list of positive and negative features, respectively. It was found that the number of features classified as *positive* is more than the number of features classified as *negative* since the domain taken for experimental purpose indicated reviewers expressing high satisfaction on the positive features. Only 5 features were identified in the negative list but that too with very low negative values as compared with their positive counterparts.

Table 3: A partial list of positive features in decreasing order of their weights

Rank	Features	Positive polarity values
1.	Picture	571.00
2.	LCD	373.86
3.	Zoom	94.75
4.	Lens	79.25
5.	Photos	46.25
6.	Color	34.26
7.	Flash	21.12
8.	Size	12.45

Table 4: A partial list of negative features in decreasing order of their weights

Rank	Features	Negative polarity values
1.	Price	-35.50
2.	Battery Life	-20.00
3.	Weight	-13.00
4.	Wide Angle	-7.60
5.	Processing Time	-1.75

3.6 Pattern Cloud Visualization

In this section, we present a feature cloud generation and visualization technique which can be used for the cases where a traditional visualization method fails to give the complete display of various information and its related concepts. The novelty of the method lies in its font generation method and visualization scheme which facilitates knowledge users to perceive mined information easily. The whole idea behind this visualization method is to connect opinions with their features in such a way that the feature having a higher polarity score is shown in a bigger font than the ones that are having lesser polarity scores. Two things are required to be taken care of while generating the visual diagram - opinions and their frequency of occurrence. Since we have already segregated the features on which the users have given their positive or negative opinions, we require generating the diagram separately for both of them. The only input required from the users is to give the maximum and minimum font-size to be used for displaying the features in the cloud. Though a number of works have been done on feature mining and visualization, to the best of our knowledge, no one has attempted to club opinion mining with feature-cloud based techniques for feature summarization and visualization.

The visualization graph has been generated to follow a star topology where the central node is the product name with which the review documents are associated. Each feasible feature extracted from the corpus constitutes a node at first outer level and linked with the central node. Finally, the set of opinions expressed over a particular feature represents individual nodes and are linked with the respective feature. Font size generation has been applied at both features and opinions to express their degree of importance with respect to the given corpus. A sample feature cloud generated from the documents related to digital camera domain is shown in figure 2. The input of this feature cloud is the data which appears in table 5 where we consider only f_i (feature), number of sentences, and w_{fi} (weight of feature f_i). The diagram has been generated considering the features at two levels of specificity. At the first level, only the features have been taken into consideration along with the number of reviews (W_{pi}) for each of these features. The font size has been calculated by taking the features and their corresponding weights. The value of maximum font size (F_{max}) and minimum font size (F_{min}) for the features has been set to 36 and 12, respectively. At the second level, we have considered the opinions related to each feature, and the feature diagram has been generated considering only the top-5 opinions related to each feature. The font size used for the case of opinions using equation 11 for each feature with the value of F_{max} and F_{min} has been taken as 24 and 10, respectively.

$$F(p_i) = (F_{max} - F_{min}) \times \left(\frac{W(p_i) - \min\{W(p_j)\}}{\max\{W(p_j)\} - \min\{W(p_j)\}} \right) + F_{min} \tag{11}$$

where F_{max} and F_{min} represent the maximum and minimum font size supplied by the users [14] and W(pi) stands for the weight of the features.

Table 5: A partial list of features along with their weight and font size for visualization

Sl.No.	Features	W(pi)	F(pi)
1.	Pictures	320	36
2.	LCD	102	15
3.	Zoom	138	19
4.	Photos	111	16
5.	Color	66	12

Table 5 provides a partial list of features along with their weights and font size values, whereas table 6 provides the opinion-wise weights of the features and their respective font size, which has been calculated using equation 4.2. The feature cloud diagram generated using the value of tables 5 and 6 is shown in figure 2.

It can be observed in the feature cloud that the terms with larger fonts are visualized easier than those that are comparatively smaller due to their enlarged font size. This effect facilitates users to realize the relevance of incorporating visualization tool with text mining systems to highlight the terms in the proportion of their importance and directly moving inside the dataset for a detailed evaluation of the reviews. Moreover, the size of the cloud can be further expatiated to accommodate more features into it, instead of only five, to have a broader overview of the content or it can be set proportional to the depth of review which can be required at different levels of specificity.

Table 6: Opinion-wise weights of the features and their font size

Sl.No.	Features	Opinion	W(pi)	F(pi)
	LCD	BIG	12	11
	LCD	BRIGHT	10	10
1	LCD	EXCELLENT	30	24
	LCD	VERY GOOD	28	23
	LCD	GOOD	22	18
	ZOOM	EXCELLENT	35	24
	ZOOM	EXCEPTIONAL	29	20
2	ZOOM	AWESOME	24	17
	ZOOM	GOOD	14	10
	ZOOM	AMAZING	36	25
	PICTURE	EXCELLENT	95	24
	PICTURE	AWESOME	65	17
3	PICTURE	AMAZING	45	13
	PICTURE	VERY GOOD	83	21
	PICTURE	GREAT	32	10
	PHOTOS	GREAT	17	10
	PHOTOS	EXCELLENT	41	24
4	PHOTOS	PERFECT	16	9
	PHOTOS	AWESOME	17	10
	PHOTOS	OUTSTANDING	20	12
	COLOR	GREAT	15	14
	COLOR	BRIGHT	12	12
5	COLOR	BEAUTIFUL	31	24
	COLOR	NICE	8	10

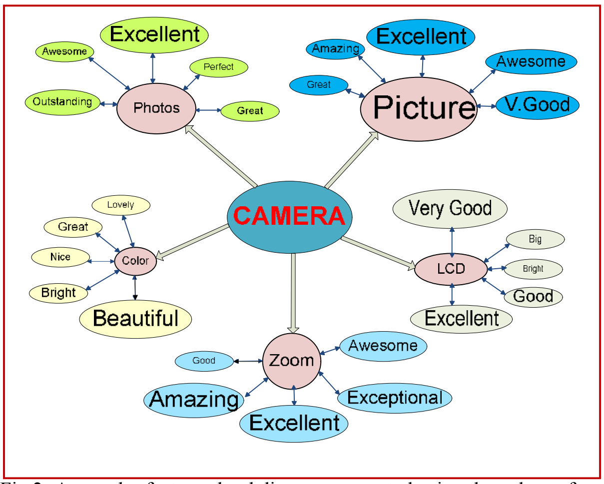

Fig 2: A sampler feature cloud diagram generated using the values of table 5 and 6

4. Conclusion

In this paper, a feature weighting mechanism has been presented that exploits expressed opinion over the features to determine the rank of the features in their respective categories. The paper also proposes an amalgamation of a novel feature cloud generation and visualization mechanisms and implemented for the purpose of visualization of features to perceive the features and related opinions without exploring the pile of review documents. As far as our knowledge of various visualization techniques is concerned in the area of opinion mining we have not come across any paper which had used feature cloud visualization technique in the gamut of opinion mining.

References

[1] M.E. Maron, and J.L. Kuhns, "On Relevance, Probabilistic Indexing and Information Retrieval", Journal of the ACM, 7(3), pp. 216–244.

[2] G. Koutrika, Z.M. Zadeh, H. Garcia-Molina, "Data Clouds: Summarizing Keyword Search Results over Structured Data", In Proceedings International Conference on Extending Database Technology (EDBT 2009), pp. 391–402.

[3] B.Y. Kuo, T. Hentrich, B.M. Good, and M.D. Wilkinson, "Tag Clouds for Summarizing Web Search Results". In Proceedings of World Wide Web 2007. ACM Press (2007), pp.1203-1204.

[4] P. Venetis, G. Koutrika, and H. Garcia-Molina, "On the Selection of Tags for Tag Clouds". In WSDM, 2011, pp. 835–844.

[5] X. Ding, B. Liu, and S.Y. Philip, "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, 2008, pp. 231-240.

[6] S. Kim, and E. Hovy, "Determining the Sentiment of Opinions", In Proceedings of the 20th International Conference on Computational Linguistics (COLING'04), Switzerland, 2004, pp. 1367-1373.

[7] A. M. Popescu, and O. Etzioni: "Extracting Product Features and Opinions from Reviews", In Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP'05), Canada, 2005, pp. 339 – 346.

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

[9] B. Pang, L. Lee, and S.Vaithyanathan, "Thumbs up? Sentiment Classification Using Machine Learning Techniques", In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP'02), USA, 2002, pp. 79 – 86.

[10] D. Grabner, M. Zanker, G. Fliedl and M. Fuchs, "Classification of Customer Reviews based on Sentiment Analysis", In 19th Conference on Information and Communication Technology in Tourism, Springer, Sweden, 2012.

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

[12] T. Ahmad and M.N. Doja, "Opinion Mining using Frequent Growth Method from Unstructured Texts", IEEE 2013 International Symposium on Computational and Business Intelligence, ISCBI 2013, pp. 92-95.

[13] G. Holmes, A. Donkin and I.H. Witten, "Weka: A Machine Learning Workbench". In Proceedings of Second Australia and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia.

[14] M. Abulaish and T. Anwar, "A Keyphrase-Based Tag Cloud Generation Framework to Conceptualize Text Documents", International Journal of Adaptive, Resilient and Autonomic Systems (IJARAS), Vol. 4, Issue 2. 2013. pp. 72-91.

[15] B. Pang, and L. Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts", In Proceedings of ACL 2004, pp. 271-278.

Tanvir Ahmad is an Associate Professor in the Department of Computer Engineering, Jamia Millia Islamia (A Central University), New Delhi. He has done his B.E. (Computer Engineering) from Bangalore University and M.Tech (Information Technology) from I.P University, New Delhi in 2006. He has submitted his Ph.D. in July 2013, from Jamia Millia Islamia in the area of Frequent and Sequential pattern mining. He started his teaching career from University Polytechnic, Jamia Millia Islamia where he joined as Lecturer in 2002. Later he joined the Department of Computer Engineering, J.M.I. in the year 2004 as a Lecturer and was promoted to the post of Reader in 2006. In 2009 he was promoted to the post of Associate Professor. He has published about 14 papers in International journals and various good conferences of IEEE, Springer and Elsevier. His area of interest includes Text Mining, Sequential Mining, Image based Security and Fuzzy Database. He is a member of IEEE and Computer Society of IEEE, ISTE and CSI.