CUSTOMER REVIEW SUMMARIZATION USING SUPERVISED LEARNING TECHNIQUES

Dissertation Report
Submitted
By

Swapnil Vijayvargiya

To

Department of Computer Science and Technology

In partial fulfilment of the Requirement for the
Award of the Degree of

Master of Technology in Computer Science

Under the guidance of

Mrs. Darvinder Kaur
Asst. Professor

(April 2015)

PAC Approval

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

School of: Computer Science and Engineering.

DISSERTATION TOPIC APPROVAL PERFORMA

Name of the Student: Swapnil VijayVargiya

Batch: 2010 - 2015

Session: 2014-2015

Registration No: 11006616

Roll No. B 31

Parent Section: K2005

Details of Supervisor:

Name: Darvinder Kaur

U.ID: 14814

Designation: Asst Prof.

Qualification: M.E.(CSE)

Research Experience: NLP

SPECIALIZATION AREA: NLP (pick from list of provided specialization areas by DAA)

PROPOSED TOPICS

1. Machine Translation (English-Punjabi Language Pair)
2. Gait Recognition
3. Image Processing

1/9/14 Signature of Supervisor 14814

PAC Remarks:

Topic 1 is approved.

APPROVAL OF PAC CHAIRPERSON: Signature: Date:

*Supervisor should finally encircle one topic out of three proposed topics and put up for approval before Project Approval Committee (PAC)

*Original copy of this format after PAC approval will be retained by the student and must be attached in the Project/Dissertation final report.

*One copy to be submitted to Supervisor.

Abstract

Sentiment Analysis is relatively a new subject of research and is beneficial in many fields. With tremendous amount of textual data over web, sentiment analysis can be easily applied and general public will have great amount of benefits from it. As opinions are the centre of everyone's decision, this topic has a great scope ahead. This report introduces a novel approach of summarizing customer's reviews about a particular product using feature based opinion mining and supervised learning techniques. Main focus in this work is on considering nouns and verbs as opinion words along with adjectives and also to determine the strength of an opinion. This work also focuses on semantics of sentence to find some implicit sentences with opinions. Reviews from Amazon [28] are used as training data. This report also contains details about some preliminary steps to be followed in sentiment analysis to improve the final results.

CERTIFICATE

This is to certify that **Swapnil Vijayvargiya** has completed M.Tech dissertation titled **Customer Review Summarization Using Supervised Learning Techniques** under my guidance and supervision. To the best of my knowledge, the present work is the result of his original investigation and study. No part of the dissertation proposal has ever been submitted for any other degree or diploma. The dissertation proposal is fit for the submission and the partial fulfilment of the conditions for the award of M.Tech Computer Science & Engineering.

Date:

Signature of Advisor

Name: Darvinder Kaur

UID:

Acknowledgement

I would like to gratefully and sincerely thank **Mrs. Darvinder Kaur** for her guidance, understanding and patience during my dissertation-I studies at Lovely Professional University. Her mentorship was paramount in providing a well-rounded experience considering my long-term career goals. She encouraged me to not only grow as an experimentalist but also as an independent thinker. I am not sure many graduate students are given the opportunity to develop their own individuality and self-sufficiency by being allowed to work with such independence. I would also like to thank **Ravinder Singh** for his help in collecting data and generating the dataset for my research and also for his valuable feedbacks throughout research.

DECLARATION

I hereby declare that the dissertation entitled, **"Customer Review Summarization Using Supervised Learning Techniques"** submitted for the M.Tech Degree is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree or diploma.

Date:

Swapnil Vijayvargiya
Reg. No. 11006616

Table of Contents

List of Tables

List of Figures

Chapter 1
INTRODUCTION

Today a tremendous amount of textual data on the World Wide Web is available, which gives a huge opportunity to apply the knowledge of Natural Language Processing (NLP) and Artificial Intelligence (AI) on this data and get some beneficial results. There are different fields in NLP like text summarization, text classification, text correction, text identification etc. All of them are really helpful and are enrich with many untouched key components for the research purpose.

1.1 Sentiment Analysis

Whenever an individual or an organization wants to make a decision they tend to seek for the opinions of other individuals and organizations. This is the reason behind the success of consultancy firms. Center of every single decision, a human being make, is opinions, sentiments and experience of other human beings. Sentiment analysis is the study of opinions and all of its synonyms like emotions, sentiments, experience, evaluation etc.

Sentiment analysis is all about finding out the views of a particular person or a group of individuals or sometimes of an organization over any object, event or even another individual or organization by analyzing the text written by them. This particular topic was not very popular in the field of research till the evolution of internet and the introduction of social networking and e-commerce websites. Reason behind the lack of interest over sentiment analysis in the past, was the lack of textual data. It is obvious that before internet or at least it can be said that before the introduction of social networking sites, e-commerce sites and online forums, if anybody needed any opinion on any particular topic he/she would have asked his/her family members, neighbors, colleagues or friends for the same, but same is not the case now. Now if someone wants an opinion on any particular thing he/she will just ask someone over any forum or social networking sites like Twitter or you will simply check the already posted reviews of individuals on the same topic or object. With internet one has the opinions of immensely large no. of people but as everyone knows everything has its own pros and cons, and same is the case with internet. The amount of data over internet is huge so it will be much more helpful and beneficial than the reviews of a small group of people in one's neighborhood but at the same time it won't be easy for someone to read out whole amount of reviews given by general public over the topic of his/her concern, and there comes the sentiment analysis into play. Data over internet is better in both quality and quantity so by summarizing that data one can take much better decisions. With the help of automated sentiment analysis tools a summarization can be done of those reviews into small data with beautiful representation in more understandable and readable form. And since other's opinions plays a very important and huge role in general public's daily lives, whether it is about watching a movie, voting for a political person/party or buying a product, everyone needs reviews and opinions, so sentiment analysis has a great scope now a days.

1.1.1 Types of Opinion Mining [26]

Opinion mining can be classified on the level on which it is applied. Basically there are three types of opinion mining on level basis:

Document Level Mining

In document level mining object remains to summarize a whole document written about a particular organization, person, product etc. as positive, negative or neutral. But the problem in document level mining is that it categorizes a whole document into positive or negative categories, but a document can't be fully positive or fully negative about the topic it is about. It is possible that some of the features or the components of the topic mentioned in the document are good (positive) and others are not so good (negative). So no one can judge an individual, product or organization on the basis of document level analysis.

Sentence Level Mining

Unlike document level mining, in sentence level mining small sentences are considered for the opinions. This is better than the document level mining but still a sentence can be comparative, superlative, conjunctive or complex sentence, which can have more than one view over different features of an object. So there must exist one more layer to go down where comes the feature based mining.

Feature Based Mining

Feature based mining gives capability to exploit the reviews and opinions for individual features of an object which will give much better results and a clear view over the object under analysis.

1.1.2 Similarity with text classification

Opinion mining or sentiment analysis is often viewed as text classification but it is different, it's more like text summarization than text classification. In text classification documents or texts are classified in different predefined classes like politics, sports, entertainment etc. But in opinion mining the concern is all about the views of public about a particular object. In text classification class related words acts as a main deal, on the other hand in sentiment analysis, words which can express sentiments of a person are the main focal point, words like good, bad, awesome, pathetic etc.

1.2 Model of Sentiment Analysis [26]

Generally, opinions are expressed on anything, e.g., a product, a service, a topic, an individual, an organization, or an event. Term used to denote the entity which is under observation, is "Object". This object have different components and attributes and then each component may have further sub-components. Thus objects are hierarchal in nature.

1.2.1 Object

Object is a hierarchy of components and attributes. An object defines an entity that can be anything from a person to an organization or from a product to a particular political party. Then in every object each component may have its own sub component and attributes, that's why objects are hierarchal in nature. But since it can be way too more complicated for general public, a new word is defined i.e. "feature" instead of "component/attribute". Since "feature" is more general word than "component/attribute" it will be easy to relate for the users and also "feature" is a word that can be related to products, for which whole summarization is done. The point to notice here is that according to definition object itself is a feature and acts as the root of the hierarchal tree.

1.2.2 Passage of Opinion

Opinion passage is the part of document that expresses the positivity or negativity about an object or a feature of a product. It is possible that more than one sentence combined can express opinions about a single feature and also sometimes a single sentence can express emotion for multiple features of a product. So passage of opinion acts as a database for a sentiment analysis system.

1.2.3 Opinion holder

Holder of opinion is the person or the organization that published that opinion. In case of this study, holders will be the authors of the reviews, comments and posts on the input website. Though in this study opinion holders are not that much important but they have their importance in some specific areas like news articles etc.

1.2.4 Semantic Orientation of Opinions

The semantic orientation of an opinion on a feature states, whether the opinion is positive, negative or neutral.
A model can be generated by combining all the definitions defined above. This new model is known as feature based opinion mining model.

1.3 Feature Based Opinion Mining [26]

As mentioned earlier, in feature based mining relevant sentences are the choice from which a large amount of data is formed and then the aim remains to extract the basic keywords related to the features of the object under observation and then finally the summary of opinions about those features is generated. On the basis of model presented earlier, the three key tasks to be performed in feature based opinion mining are:

Identifying object features: For example, in the sentence "This phone has a great camera and motion sensor." the object features are "camera quality" and "motion sensor". A supervised pattern mining method is proposed in "*Opinion Observer*" [1], to identify the features. An unsupervised technique is implemented in "*Mining and Summarizing Customer Reviews*" [2]. In "*Opinion Observer*" [1] their technique basically finds frequent nouns and noun phrases as features, which are usually genuine features. But as the main goal of this work says noun and noun phrases alone are not enough, for much more accuracy some infrequent features must be extracted as done in Hu and Liu [2]. There are many more information extraction techniques available, for instance, CRF is one then there is hidden Markov model (HMM), and the list goes on. In this work frequent features are already defined for mobile phones and to perform mining on objects other than a mobile phone, one can either introduce the list of frequent features (recommended for more accuracy) or let the system to extract the features by itself.

Determining Opinion Orientation: Task here is to determine the orientation (positive, negative or neutral) of the sentence. For instance, example covered in previous paragraph has the positive opinion about both the given features. For this task also there are many techniques available. A lexicon-based technique has been implemented in *"Holistic Lexicon-Based Approach to Opinion Mining"* **[3]** and it gave quite a good result. Lexicon-Based techniques uses opinion words to determine the orientation, and same is the idea for this work also. In this work pre-defined opinion words are be used which will be defined in word-orientation pair. Then there are other supervised learning techniques that can be used.

Grouping Synonyms: There can be many words that can express same opinion for a particular feature of the object, task here is to group those words and the term for these kind of words is synonyms. An attempt has been made in "*Extracting Knowledge from evaluating text*" **[4]**. In this work since the list of opinion words is already provided hence this part of feature based mining is excluded.

1.4 Evolution of Opinion Mining[27]

Right now Opinion mining is all about extracting the most important text features. Two most commonly used features are Presence and Term-Frequency. Presence is just a binary value which shows whether a particular text is present in document or not. It is same as finding a sub-string in a string. It is more effective for reviewing polarity classification. It also shows that recurrent keywords may indicate a topic, but that doesn't mean that repeated terms have something to do with overall sentiment.

Another possible feature is position. Position shows that how a particular word can differ the sentiment of overall sentence by shifting its position in the sentence. Sometimes, n-grams are also considered as important features.

Some techniques also look into the distance between different words (Opinion words and product features). Generally, part of speech (POS) information (for example, nouns, adjectives, adverbs, and verbs) is used for word-sense disambiguation.

Some researchers have developed other techniques orientation to pre-defined classes to determine the polarity. These types of approaches are used in document level analysis and are strictly reserved to a particular domain or topic.

1.4.1 Heuristics vs. Discourse Structure

Some of the unsupervised learning approaches generates a sentiment lexicon which is further used to determine orientation. This is a crucial step in many summarizing techniques. It defines the prior orientation of a term or a phrase which helps to identify the contextual polarity afterwards. Earlier researchers focused on linguistic heuristics only but by doing so they were unable to define the real sentiments so now a days researches are focused seed words with predefined orientation. Sometimes, like in this work strength of opinions is also calculated. For such calculations regression is used. These techniques looks at the similarity between two or more words and by looking at the relationship between those words a point on a scale is generated to determine the strength of review.

Discourse Structure means the twists and turns in a document. By dealing with such structures one can increase the accuracy in labelling the strength of sentiments. Researchers dealt with this problem by looking at the position of a particular sentence in a document. Generally the last sentences of documents, conclude it and they have more accurate and strong opinion about the document than the rest of the text.

1.4.2 Keywords vs. concepts

Evolution of sentiment analysis can be described by looking at the approaches used to summarize a text. Current approaches can be classified into following categories:

Keyword spotting: Keyword spotting is an approach where the system looks out for a general opinion word like happy, sad, excellent etc. and then tries to identify the opinion of whole sentence. A modified version of this approach is used in this work. Drawback of this approach is that it can't reliably find out the negated words. Using keyword spotting a system will correctly classify *"I am happy"* as a positive sentence because of the word *"happy"* but it will also classify *"I am not happy"* in the same category. Because it will ignore the negated word *"not"* and classify the sentence on the basis of word *"happy"*.

Second drawback of this approach is that it relies on the fact that each opinion sentence will contain at least one opinion word (adjective) but sometimes a sentence can show sentiments without the use of an adjective. For example, "*I filed a copyright infringement case against him.*" Doesn't contain any opinion word but has a very strong emotion.

Lexical affinity: This technique also detects the opinion words but apart from that it also assigns a probable affinity to each word. For example word *"sucks"* can have a 90% affinity to show a negative emotion. For example in, *"Camera of this phone really sucks"*, but same is not the case in *"Your vacuum cleaner sucks nicely"*, here *"sucks"* defines a positive emotion and hence it has affinity of 90% and not 100%. Affinity in these type of techniques are trained using linguistic database. This technique is clearly better than keyword spotting but has its own cons. Two major drawbacks are:

Negated sentences and sentences with some other meanings fails lexical affinity because of their operation on word level only.

Affinities are generally biased depending upon the genre of the linguistic database and hence domain-independent model can't be constructed.

Statistical methods: Statistical methods involves machine learning techniques like Bayesian theorem and SVM, and are proved to be very good in text classification. Statistical methods are implemented by providing a machine learning algorithm and a linguistic corpus to the system. And system itself learns to identify the opinion words as well as the orientation of those opinion words. So basically it serves the objectives of both keyword spotting and lexical affinity methods. Drawback of statistical methods are that they need a large text input data to learn neatly and efficiently. Which basically means that these types of techniques can be used to do document level or page level mining but can't be used in sentence level or feature based mining.

Concept-based approaches: These techniques use semantic networks to perform semantic analysis and are way better than the syntactic analysis methods described above. These types of methods gives us more semantic results by looking into a huge semantic knowledgebase. So at one hand system provides better results because now it doesn't trust blindly on some opinion words or on no. of occurrence of a word or a phrase but on the other hand developer has to provide a huge knowledge base with all the semantic meaning that are needed to perform the analysis. Without a well-defined knowledge base a concept based system may perform poorer than its syntactic counterparts.

1.4.3 Multimodal Sentiment Analysis

With evolution of internet there are many new sources of describing opinions are available. People are no more bounded to the textual data. They can now post a photo describing their emotion on twitter, Facebook or instagram or they can even post a video about their opinion. There are no boundaries when it comes to sources of connecting to other peoples. So there must be some new technologies that can summarize the visual and audio data along with the text data. This type of techniques are known as multimodal sentiment analysis or opinion mining. Currently no research completely focused on multimodal sentiment analysis is going on. This part of sentiment analysis is yet to be explored but ones it unfolds it has great hidden applications.

Chapter 2
Literature Review

Despite the fact mentioned in previous chapter that researchers focused themselves in sentiment analysis recently, there has been a tremendous amount of work done in the field. Some of them were trying to summarize an event like a football game and some of them were focused in summarizing the reviews for a product or movie.

DeJong [9] gave an overview of FRUMP system in his book. This program was developed at Yale Artificial Intelligence project. It was one of the best artificial intelligent program in the field at that time. And the reason behind this appreciation is the diversity of input it can handle. This system can generate report on any topic from earthquakes to forest fires and wars. That program had sketchy scripts for 60 different situations.

Tait [10] have worked in template instantiation, a subsidiary of text summarization. His project was not domain independent since he had to extract certain facts and entities from a document that was packaged in a template. To obtain this goal he needed some prior knowledge that can instantiate the template to a certain level of details. The program was based on technology similar to a script applier. However, that program has advance results as compare to the previous works in the field. It has better results than previous programs and needed less information about the domain of the input. Following are the key ideas of his project that gave him such good results.
The very first thing he did was to incorporate an automated algorithm that can extract information about input instead of asking it from user.
Next big thing was the allowance of more than one topic in a section. And his algorithm was good enough to process and summarize these topics simultaneously.
And lastly it incorporates a mechanism to copy some unexpected text from input to output keeping in mind that these unexpected texts can be the key text in input and will demolish the objective and semantic of whole text if omitted in the output.

Works of Hearst [13] and Sack [14] on sentiment-based classification of entire documents use models inspired by cognitive linguistics. Sack mainly worked on point of view. According to him system to summarize text of stories and news does not take into consideration the way in which the story is told. And that makes almost every system at that time a “gullible reader” [20]. His techniques can be used in the entertainment industry to find and then assemble different text and video clips with same point of view to tell a story.

Boguraev and Kennedy [11] proposed to summarize a whole document on the basis of some key entities, events or expressions. Their work was a domain independent project, and does not require an in-depth analysis of the full meaning. But despite this feature it remain close to the core meaning by providing different forms of its representations. Work in this report is different from their work because they only focused on some important factors but in this work objective is to find out all the features whether they are prominent or not.

Huettner and Subasic [16] combined lexicon method with fuzzy logic to achieve their aim. They combined NLP and Fuzzy Logic to generate a system for document analysis and management. They used semantic typing in their project. And they used fuzzy typing for analyzing affect. Fuzzy typing consists following concepts:

- Isolating a vocabulary of words belonging to a metalinguistic domain.
- Using multiple categorizations and scalar metrics to represent the meaning of each word in that domain.
- Computing profiles for texts based on the categorizations and scores of their component domain words.
- Manipulating the profiles to categorize, differentiate, cluster, match, or visualize the texts.

They build their lexicon using 4000 English words covering almost each domain. And then there are some ambiguous words, to deal with them they simply assign the words to each domain which make sense with meaning of those words.

Das and Chen [15] tried to classify the stock postings on an investor bulletin by combining statistics and lexicon method.

Tong [7] developed a timeline for a movie that tells that at a particular time in the movie what most people are talking about. So it shows the opinions and sentiments of public at different parts of the movie along with an overall review of full movie. Sentiment for a particular part is defined by summarizing the comments of different people on that particular part. And as the definition of lexicon-based techniques, which is discussed in previous section, this system falls into same category i.e. some predefined words were used to classify the messages.

Bo Pang, Lillian Lee and Vaithyanathan [8] used machine learning techniques to classify movies into positive and negative reviews. Their work was on sentiment classification and the techniques they used were Naïve-Bayes' classifier, maximum entropy classification and support vector machine. The results showed that these techniques can be used for sentiment analysis but are better if used for traditional topic based classification. They used IMDB's newsgroup as their database. Unlike other sentiment classification works, they didn't classify documents on the basis of some predefined classes. Instead they gave a full review of a whole document that is the deviation of document towards positive or negative side. According to their experiments and results their system works better than the human baseline classifier and has accuracy of up to 82%.

Dave et al. [25] describes a tool for shifting through and synthesizing product reviews, automating the sort of work done by aggregation sites or clipping services. They began by using structured reviews for testing and training, identifying appropriate features and scoring methods from information retrieval for determining the polarity of the reviews. Results using this technique were same if not better as acquired by similar projects using machine learning techniques. Then they used their classifier to work on internet to classify reviews there which is a more complex work space and the work of classification there is harder. But they used a simple technique that to identify the relevant features of a product and that resulted into a useful summary.

Bing Liu and Hu [2] did a work same like this one, but their approach only considered adjectives as opinion words and also they used WordNet to determine the orientation of their opinion word and in this work an efficient corpus based technique is used instead, which gives better results with less complexity. One can also fetch orientation of any word by calculating the mutual information after firing a query to a search engine with the words "excellent" and "poor", but then again it is way too more complex to fire a search query for each and every word. In their work they also left out to find out the strength of opinions, which is one of the task in this work.

Popescu et al. [19] developed an unsupervised information extraction system called OPINE, which extracted features of products and identify opinions about those features. It considers some threshold frequency value to limit the extracted features. This threshold was defined by considering the results and conclusions of several experiments done before. After taking into consideration the remaining feature the assessor of OPINE finds out some explicit features. It evaluates a noun phrase by computing a Point-wise Mutual Information score between the phrase and metonymy discriminators associated with the product class.

This work also uses a similar idea to extract the features but that is the case when list of features is not provided by the user. The main approach is to ask the user for the list of features, it will give better results. A list for features of mobile is already included within the system. One needs to provide the list if he/she is performing analysis on some other product than a mobile phone.

Abulaish et al. [20] presents an opinion mining system that can identify product features and opinions from documents instead of internet. Semantic analysis was done to extract these features and opinions. To determine the deviation of opinions they used a polarity score of opinion words through WordNet and then they generated a summary for the whole document. They also generated a visualization report to provide a better representation of the summary generated.

Parikh and Movassate [21] created a program to analyze the different tweets. Their program was target independent and instead of giving opinion and summary about a particular product or movie they generated summary for each and every single tweet they came across. They used machine learning techniques such as some modified versions of Naïve Bayes classifier and Maximum Entropy Model. They had 370 tweets each of both negative and positive sentiments. They used 100 of them from each class for testing purpose and rest of them were used for the training purpose. They also used a java archive called "jttwitter.jar", which is a twitter API specially built for Java. They used such small set of data because according to them, the hardest part in twitter sentiment analysis is to collect the data and it will take ample amount of time so they decided to go with small set. The reason they gave behind the difficulty in collecting data from tweeter is that most of the tweets contains only a link or else they are written in some foreign language.

Go et al. [24] also worked on twitter sentiment analysis and they chose to work with machine learning techniques like Naïve Bayes, Maximum Entropy and SVM. They also proved that by using twitter emoticons (smileys) as training data these techniques can provide accurate results up to 80 %. So they also used emoticons as their training data same as Davidov et al. [23]. They also gave some preprocessing steps that can help into improving the accuracy and efficiency of a sentiment analysis system. They used supervised learning for their system. They used some predefined twitter API's to extract the tweets with emoticons for their training data and same as Davidov et al. [23] they also avoided the extensive work of handcrafting manual data for training and testing consisting of some opinion determining words.

Barbosa and Feng [22] also worked on twitter sentiment analysis. Since twitter has lots of noise in its tweets because of the character limit of 140 characters, they chose to collect data from different sentiment detection websites instead of manually collecting data directly from twitter. This data was noisy data and helped them to build a more robust system then the existing ones. They used a two-way approach which classifies each tweet firstly into subjective and objective tweets, and then in second step they mark the subjective tweets with the label of positivity and negativity. This approach was similar to that of Pang and Lee [8].

Davidov et al. [23] also worked on twitter sentiment analysis, but their work was different than any other approaches and works discussed till now. Instead of using opinion words like any other work they used twitter hashtags and smileys to figure out the polarity of a tweet. This way they build an efficient system and also avoids the labor of manual detection of the opinion words. They used 50 different hashtags and 15 smileys for their work.

Jiang and Yu et al. [5] did a beautiful job by merging target dependent and context aware approaches together to consider the semantic meaning of sentences instead of just using the target and the sentiment defining words. This approach produced a system that can understand the meaning of sentence and can judge, whether the sentence is having some sentiments about the target mentioned in it or not. Other problem that was solved by their approach was the polarities of the sentences in which a sentence is comparative that gives a positive opinion for one target and negative for other. Following are the examples mentioned in their paper for both kind of sentences:

1. *"People everywhere love Windows & vista. Bill Gates"*
2. *"Windows 7 is much better than Vista!"*

Example 1 is neutral about the target *"Bill Gates"*, but the sentiment analysis systems at that time were classifying it as a positive sentence. And in second example, target independent systems at that time were unable to separate the different targets in comparative sentences like this. This sentence would have been resulted as a positive sentence for both the targets *"Windows 7"* and *"Vista"*. Their system worked well with an accuracy of 66-68 %.

Chakrabarti and Punera **[6]** wrote a paper on real time sentiment analysis and summarization of an ongoing event like a football game. According to them they were the first to summarize the live events using tweets. They used a two-step process to do so. Firstly their algorithm had a modified Hidden Markov Model that can segment the event time-line, depending on both the burstiness of the tweet-stream and the word distribution used in tweets. And then it picks up the key tweets to describe each segment judged to be interesting enough, and combine them together to build the summary. Separating sub events and then restricting those sub events was one of the greatest challenge of their work. Since a sub event in a real time event like a football game may create millions of tweets, so their algorithm has to restrict those tweets to a limited extent but with relevant and most heated tweets. Other challenges were to remove the noise of tweets. Since twitter gives its users a limit of 140 characters for a tweet, it is important to check those tweets for abbreviations, short hand typing and other types of noises. And then there are previous instances of similar events like some previous games between same pair of competitors, this problem can provide you tweets from earlier games and will result in wrong summary.

Bing Liu **[26]** wrote a summary of opinion mining which provides a brief history and basic concepts in the field and gives a model for sentiment analysis and feature based opinion mining. It also has a brief description about mining comparative and superlative sentences which instead of giving direct opinion on an object or about an event, gives a comparison between two or more objects and/or events.

Works discussed till now focuses on single document. But there are multiple researchers who tried to apply their knowledge of sentiment analysis on multiple documents at the same time. These documents had same background that means their topics are somehow related to each other. Generally the output of these kinds of work comes out to be a comparative summary which compares two or more documents on the basis of some common features. Mani et al. [12] is an example of that. They described a new method (at that time) for generating comparative summaries for related documents. They generate graphs from the input documents in which different words like nouns and important noun phrases acts as nodes and the semantic meaning that relates those words acts as edges between those nodes. Their idea was to first generate separate graphs for both the input documents and then they compare the similarities and differences between those graphs to generate the summary. These graphs were developed by determining the relation of the words used as nodes with the topic of the documents.

This work is different from their work because the aim of this work is to find the key features that are talked about in multiple reviews and also because in this work key task is to deal with no. of reviews about one product and not with multiple documents and also in this work products are not compared with each other, here task is to summarize reviews for a single product only.

Chapter 3

Objectives of Study

Work represented by this report is fully focused on feature based sentiment analysis and customer review summarization. This work is going to be a complement to the work done by Minqing Hu and Bing Liu [2]. They also worked on the same topic of customer review summarization but the shortcoming of their work was that they only considered adjectives as opinion words but in this work verbs, adverbs and nouns are also considered to be the opinion words. They also left calculating the strength of opinions for their future work and that part is completed in this work.

- Formulated problem by doing an extensive literature survey.
- Proposed an efficient solution for solving the problem.
- Developed a system based on the proposed solution. This system considers opinion words other than adjectives, it also calculates the strength of opinions and finally it generates the summary of the reviews.
- To test and calculate the final results of system.

Chapter 4

Scope of Study

As mentioned in previous sections that opinions are the centre of most of the decisions, public makes, hence opinion mining can be used in many practical application and it is beneficial for both individuals and organizations.

Individual consumers: While purchasing a product everyone wants to read reviews, before adding it to the cart and finalizing it, but it will be more beneficial if they have to just look up to a summary of those reviews instead of reading them all. And the best thing would be if they get to see a comparative report of the competitive products in the market on the basis of those reviews.

Organizations and businesses: Business is all about customer satisfaction. And it is not possible for a big firm and organizations to get in touch with every single customers of theirs, to get their feedbacks. So, a system like this will help them to know the feelings of consumers about their products which will result into better business planning and an enhanced business.

Sentiment analysis has its own role in politics now a days. A political party can analyse the current topics in general public discussion and the problems public is facing and then they can built their election campaign and motto around those topics. That will surely help them to win and help people. Products in this problem are to be replaced with the expected or guessed hot topics as a target and the system will be able to give us the summary.

Chapter 5

Research Methodology

Proposed system is developed using *Java 1.8* hence minimum required version of Java to run this system is 1.8. Key benefits of using Java are platform independency, security, backward compatibility, dynamic and extensible programming, object oriented programs, internationalization, efficiency and performance and also the fact that Java is a popular language so there are many open source API's available that helps and reduce the work of an programmer. Apart from java following tools are used to develop this system:

5.1 Tools Description

5.1.1 Stanford POS Tagger (Maxent Tagger) [30] [31]

A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc. Maxent Tagger supports five different languages that are: English, Chinese, German, Spanish and French. Although generally computational applications use more fine-grained POS tags like 'noun-plural'. This software is a Java implementation of the log-linear part-of-speech taggers described in [30] [31].

The tagger was originally written by Kristina Toutanova. Since that time, Dan Klein, Christopher Manning, William Morgan, Anna Rafferty, Michel Galley, and John Bauer have improved its speed, performance, usability, and support for other languages.

The system requires Java 1.8+ to be installed. Depending on whether the system is running 32 or 64 bit Java and the complexity of the tagger model, one will need somewhere between 60 and 200 MB of memory to run a trained tagger (i.e., one may need to give java an option like java -mx200m). Plenty of memory is needed to train a tagger. It again depends on the complexity of the model but at least 1GB is usually needed, often more.

Several downloads are available on Stanford NLP Group's official website [34]. The basic download contains two trained tagger models for English. The full download contains three trained English tagger models, an Arabic tagger model, a Chinese tagger model, a French tagger model, and a German tagger model. Both versions include the same source and other required files. The tagger can be retrained on any language, given POS-annotated training text for the language.

POS Name abbreviations are used by English Tagger are the one defined in Penn Treebank Project [33] tag set. The tagger is licensed under the GNU General Public License (v2 or later). Open source licensing is under the *full* GPL, which allows many free uses. For distributors of proprietary software, commercial licensing is available.

5.1.2 JFreeChart Java API [32]

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

- A consistent and well-documented API, supporting a wide range of chart types.
- A flexible design that is easy to extend, and targets both server-side and client-side applications;
- Support for many output types, including Swing and JavaFX components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);
- JFreeChart is open source or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

The JFreeChart project was founded fifteen years ago, in February 2000, by David Gilbert. Today, JFreeChart is the most widely used chart library for Java (There is a list of some of the products and projects that use JFreeChart on their official website [32]), with more than 2.2 million downloads to date. The project continues to be managed by David Gilbert, with contributions from a diverse community of developers. If you are interested in joining the project, please see the Developers page.

5.1.3 Amazon Crawler

Amazon crawler is a web crawler, especially designed and coded for this project. It basically takes an ASIN ID [29] of a particular product and crawls all the reviews mentioned for that product on Amazon [28].

To fetch the review it goes through an extensive algorithm mentioned below:

Step 1: Generate URL

First step is to generate the URL from where product reviews are to be fetched. Amazon [28] saves its reviews for a particular product on a URL generated using products ASIN ID [29].

Following is the format of URL generated by Amazon [28]:

http://www.amazon.com/product-reviews/{ASIN ID}/?showViewpoints=0&sortBy=byRankDescending&pageNumber={Page No.}

{ASIN ID} in the URL mentioned above is replaced by the ASIN ID [29] of the product whose reviews are to be fetched and since reviews are distributed over several web pages hence *{Page No.}* is replaced by a numerical value that defines the page no.

An example of such URL is as follows:

http://www.amazon.com/product-reviews/B002U28LZC/?showViewpoints=0&sortBy=byRankDescending&pageNumber=1

So the algorithm will generate a URL based on the ASIN ID [29] provided by the user and *Page No. = 1*.

Step 2: Get the source code of the webpage.

Next step is to get the source code of the webpage defined by the URL which is generated in previous step.

To do so *HTTPConnection* class of *java.net* package is used along with *BufferedReader* class of *java.io package*.

Step 3: Crawl the webpage

To crawl a webpage first of all the algorithm checks that whether there are any reviews present on that page or not. If not algorithm stops. Else it starts to look for *<span>* tags with an attribute *review-text*. Because these span tags contains the review text. After finding out all the reviews on a webpage by repeating this approach, algorithm increments the page no. in the URL and it returns to step 2.

Flow chart of the proposed is system is showed in Fig. 1.

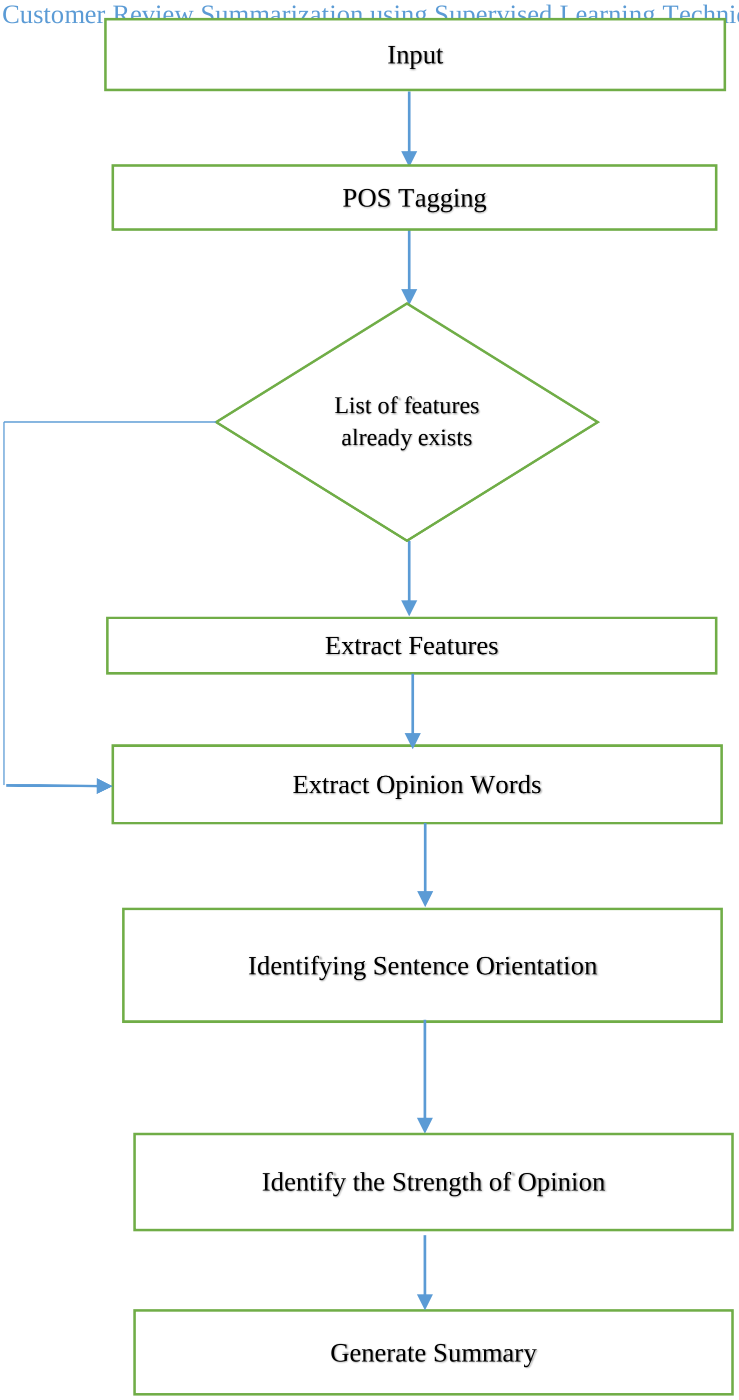


Fig. 1. Flow chart defining the working of system. (Adapted and extended version of Hu and Bing Liu [2])

5.2 Input

There are two ways to give input to this system. One is to provide a file with list of reviews which contains one review per line. Or to provide ASIN [29] ID of a product. In case of ASIN ID [29] a web crawler will crawl Amazon's webpages for reviews. It will collect top 200 reviews about the product.

5.3 POS Tagging

Product features are generally described by nouns or noun phrases. And if the aim is to deal with these features only, there must be a way to extract and differentiate these nouns and noun phrases differently. And here comes the POS tagging into role. Part of Speech tagging is done using Stanford's Maxent Tagger [30] [31]. Sentences will be saved with the tags in the database. These tags will help to find out the nouns, noun phrases, adjectives, verbs and adverbs. And other words will be discarded because there is a very little chance that those words consists a product feature or an opinion word.

5.4 Extract Features

If feature list is not provided then system has to extract frequent features from the list of reviews. It extracts feature by calculating the no. of occurrences of a noun/noun phrase in all the reviews and then by matching that counter with a particular threshold. This threshold is defined by considering the results and conclusions of several experiments done while testing phase.

To extract reviews following procedure is used:

Step 1: Create a list *"Feature List"* to hold the features.

Step 2: Generate a list *"Review List"* of all reviews.

Step 3: Select a review from the list.

Step 4: Select one noun/noun phrase from selected review.

Step 5: If *"Feature List"* doesn't contain the selected noun/noun phrase then

Add the noun/noun phrase to the list with a counter 0.

Else

Increment the counter of selected noun/noun phrase.

Step 6: If selected review has more noun/noun phrase then

Go to step 4.

Step 7: Remove selected review from *"Review List"*.

Step 8: If *"Review List"* has more reviews then

Go to step 3.

Step 9: Remove all reviews from *"Review List"* those have *count<threshold.*

Step 10: Stop.

By using this method system will only extract frequent features. But there are some infrequent features also for each products that will be ignored by this method.

Infrequent features unlike frequent ones are not the most common that one will see in the comment section for a particular product. But these features can't be just neglected. These features may play an important role in the decision of a person buying the product. And infrequent features are more important to manufacturers than to the consumers, since they have to handle every single thing in their product irrespective of the importance of that feature in the final product. Since one adjective word can be used to describe different objects, opinion words can be used to look for features that cannot be found in the frequent feature generation step using association mining. One way is to find the nearest noun/noun phrases that are effected by the opinion words. These noun/pronoun will be the infrequent words. This technique is adapted from Minqing Hu and Bing Liu **[2]**. They also discussed the problem with finding infrequent features i.e. sometimes program will get words that are irrelevant to the product. An example of infrequent feature can be the "back cover" of a mobile, or the "data cable" or even it can be the software CD that comes along with the mobile handset.

In this work no method is implemented to find infrequent feature but as mentioned these infrequent features are important for some people so this work deal with those features by combining them and consider them as one feature and that feature in this system is named as *"misc."* (Short for miscellaneous). One can look at the list of positive and negative reviews generated for *"misc."* to see the reviews about these infrequent features.

5.5 Opinion words Extraction

Next step is to identify opinion words. These are the key words that expresses the opinion about the product in a sentence. Previous work like Bruce and Wiebe's **[17, 18]** on sentiment analysis gives us positive indication that adjectives are the words that are mostly used to describe the sentiments about a particular feature. So main idea is to check all the adjectives whether they expresses an opinion or not. But as mentioned before adverbs and verbs can also act as an opinion word so in this work adjectives are not the only word to be extracted, idea is to check all the adverbs and verbs also to see if they are acting as opinion words. Though sentences to find these opinion words are restricted to the ones that have at least one product feature defined in it, since only the opinion about the product features are to be dealt with. That means this system doesn't deal with implicit sentences. To extract opinion word system will extract all the adjectives, verbs and adverbs from the list of reviews and then after matching them with the corpus, it will discard the ones with neutral orientation.

5.6 Identifying Opinion Word Orientation

Bing Liu and Hu [2] used WordNet to determine the orientation of their opinion word and in this work an efficient corpus based technique is used instead, which gives better results with less complexity. One can also fetch orientation of any word by calculating the mutual information after firing a query to a search engine with the words "excellent" and "poor", but then again it is way too more complex to fire a search query for each and every word. So in this work to find the orientation of an opinion word system has to look in the list of predefined words which are stored in word-orientation pair. Corpus used in this work consists of 2277 positive words and 5129 negative words.

5.7 Identifying Opinion Sentence Orientation

After getting the orientation of each opinion word the sentences can be classified accordingly to be a positive or negative sentence. This task can be performed by looking up at the opinion word and the feature it is describing so here basic arithmetic rules will be applied. A positive opinion with all positive words is a positive sentence. A negative opinion with negative words is also a positive sentence. A positive opinion with negative words will be a negative sentence. For ex. "*Strap is not good.*" In this sentence "*good"* is a positive opinion word but a negative word "*not*" so overall feel of the sentence is negative.

5.8 Identifying Strength of Opinion

Since orientation of review words is defined by integer value in this system one can look at that value to find out the strength of that opinion word and by combining these strengths, strength of overall review can be calculated.

5.9 Summary Generation

Each feature is saved with two different lists one for keeping positive reviews and other for negative reviews for that particular features. Summary of reviews is generated by the system with two different level of abstraction.

One is with the pie chart and bar chart showing the no. of positive and negative review for a particular feature. To generate pie charts and bar graph JFreeChart [32] API is used. Lists mentioned in the starting of this paragraph are used as the dataset for the pie chart and the bar chart.

Second is the list of positive and negative reviews for each feature separately.

Chapter 6

Results and Discussion

6.1 Result

System is tested on 5 different products with 200 reviews of each and the final results given by the system are given in Table 1. It shows the no. of different features considered for a particular product, percentage of accurate result (accuracy of system), column of false positive shows the percentage of error where the system classified a review positive when it was not positive and column of false negative shows the percentage of error when the system classified a review as negative when it was not negative. Average of these reviews gives the total accuracy of system that is 80.68%.

Product #	ASIN ID	No. of Features	Correct Result (Accuracy) (%)	Error (%)	False Positive (%)	False Negative (%)
Product 1	B00K0NRYF6	17	84.4	15.6	6.38	9.22
Product 2	B00CIF9MJK	13	80.77	19.23	12.82	6.41
Product 3	B00AXSXDFI	17	86.46	13.54	8.33	5.21
Product 4	B0097CZJEO	8	70.15	29.85	23.88	5.97
Product 5	B00HHZWO78	19	81.6	18.4	9.6	8.8
Average	**-**	**-**	**80.68**	**19.32**	**12.2**	**7.12**

Table 1. Results of final system

6.2 Output Screenshots and Description

Following are the snippets of the final system on different stages, describing the whole system.

Fig. 2. Input Choice Window

This is the very first window that comes up when one starts the system. As mentioned earlier in methodology part that this system can either take reviews from a file or from Amazon directly so this window lets user to choose between the sources of reviews.

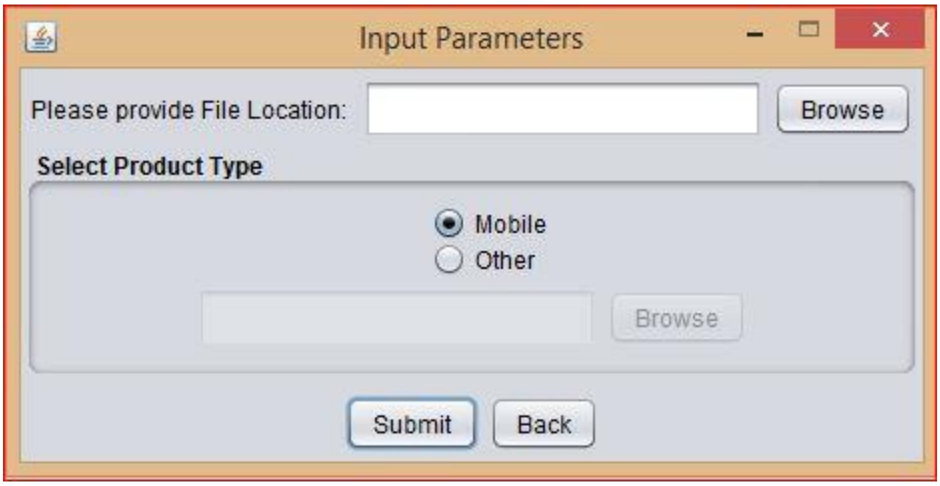


Fig. 3. Input Parameter Window (a)

This is the window which the user will face if he/she chose file as a source of reviews.

In this window user have to provide the location of the source file and he/she also has to define whether the product is a mobile phone or is of some other category.

If user choses "*Other*" as product type then the *text box* and the *browse button* in *"Product Type Panel"* will become enabled and now user has to provide another files location that contains the list of frequent features along with their keywords.

A list of feature has a pre-defined format that must be followed for proper functioning or else the system will malfunction. The format for *feature-list* is as follows:

{Keyword 1 for feature 1} {Feature 1}

{Keyword 2 for feature 1} {Feature 1}

. .

. .

. .

{Keyword n for feature 1} {Feature 1}

{Keyword 1 for feature 2} {Feature 2}

{Keyword 2 for feature 2} {Feature 2}

. .

. .

. .

{Keyword m for feature 2} {Feature 2}

. .

. .

. .

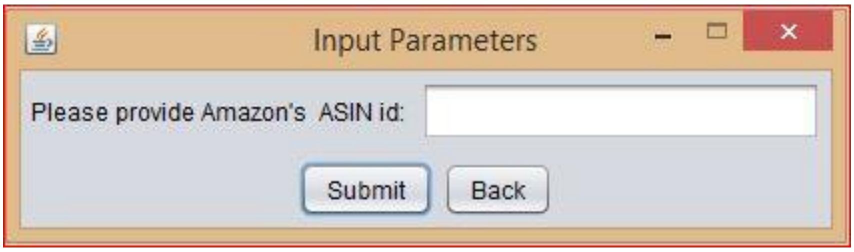


Fig. 4. Input Parameter Window (b)

This window will come up by skipping the previous one if user chose Amazon [28] as source of reviews on the very first window. It takes only the ASIN ID [29] the product for which the summary has to be generated. User can found ASIN ID [29] on product's homepage at Amazon [28] under "Product Information" heading as in Fig. 5.

Fig. 5. Snippet of Amazon's Webpage showing the location of ASIN ID [29]

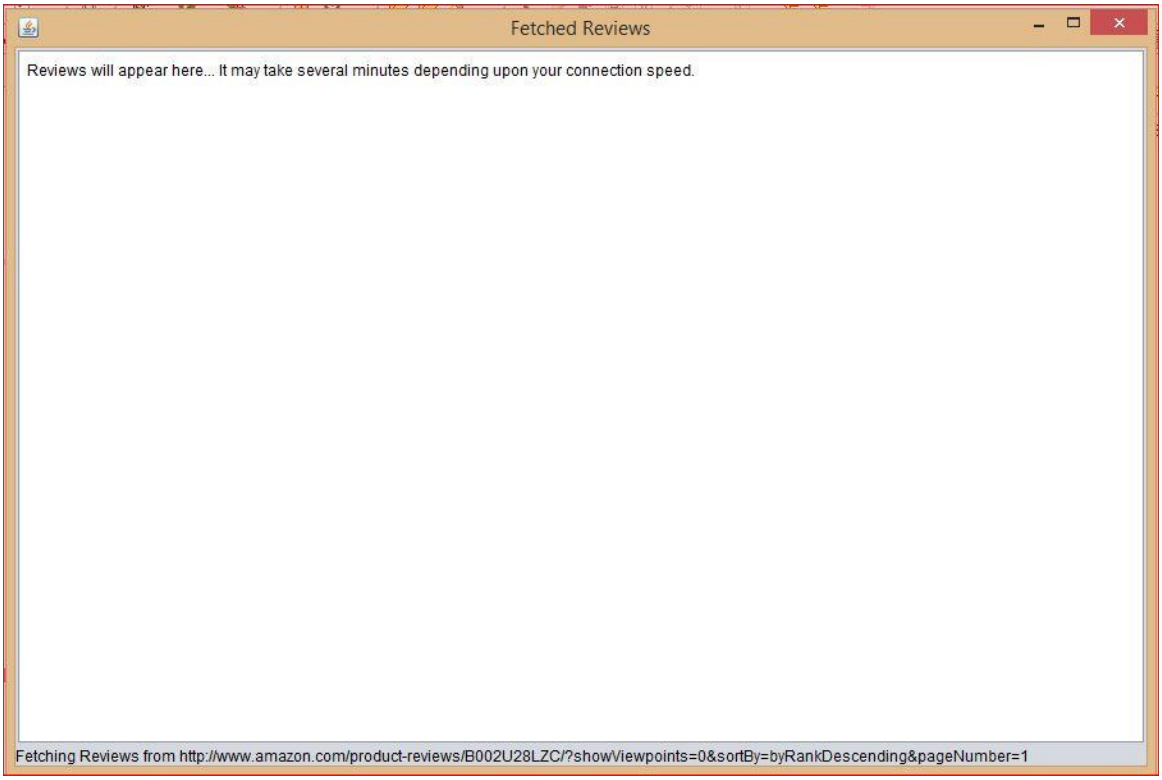


Fig. 6. System fetching reviews from Amazon

This window will come up after entering the ASIN ID [29] in previous window. It shows the webpage's URL at the bottom from which currently the reviews are being fetched. After the completion of crawling process the reviews will be displayed in this window only, as showed in Fig. 8. If user is fetching reviews from Amazon [28] then the system will first save all the reviews that are fetched by the crawler in a file in user's current directory. This file will be named according to system that and time to generate a unique name each time. And this file can be used in future as source for reviews which will be beneficial if you don't have internet connection at that time. Even if you have internet connection then also fetching reviews from file is much more efficient and quicker than to fetch them from internet.

Fig. 7 shows the window after the reviews are crawled and saved in file. It shows a message for user which describes the file location and name. Although the reviews are numbered in the window but same will not be the case with the file. Keeping in mind the format of *source file* system can't save the reviews with nos. in file. Numbering is window is just for increasing the readability for the user.

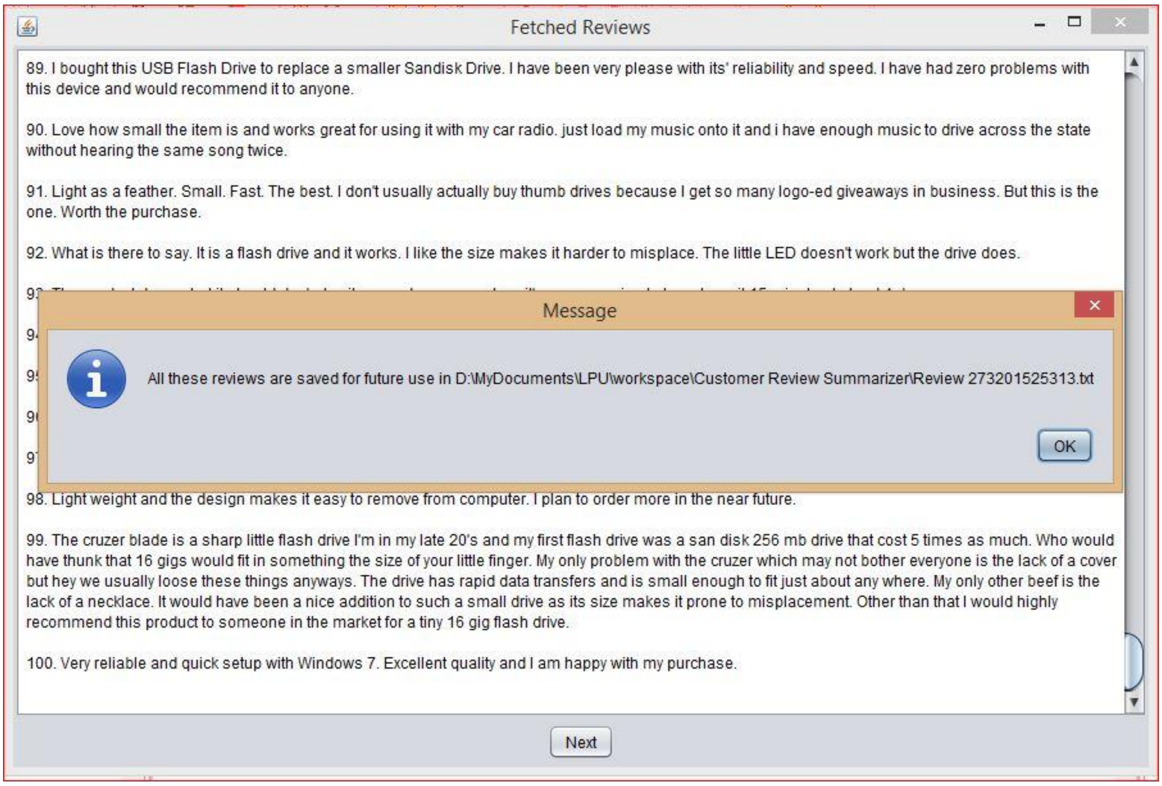


Fig. 7 Window showing the saved file location

On pressing *next button* in the window showed in Fig. 7. System will proceed for the tagging process.

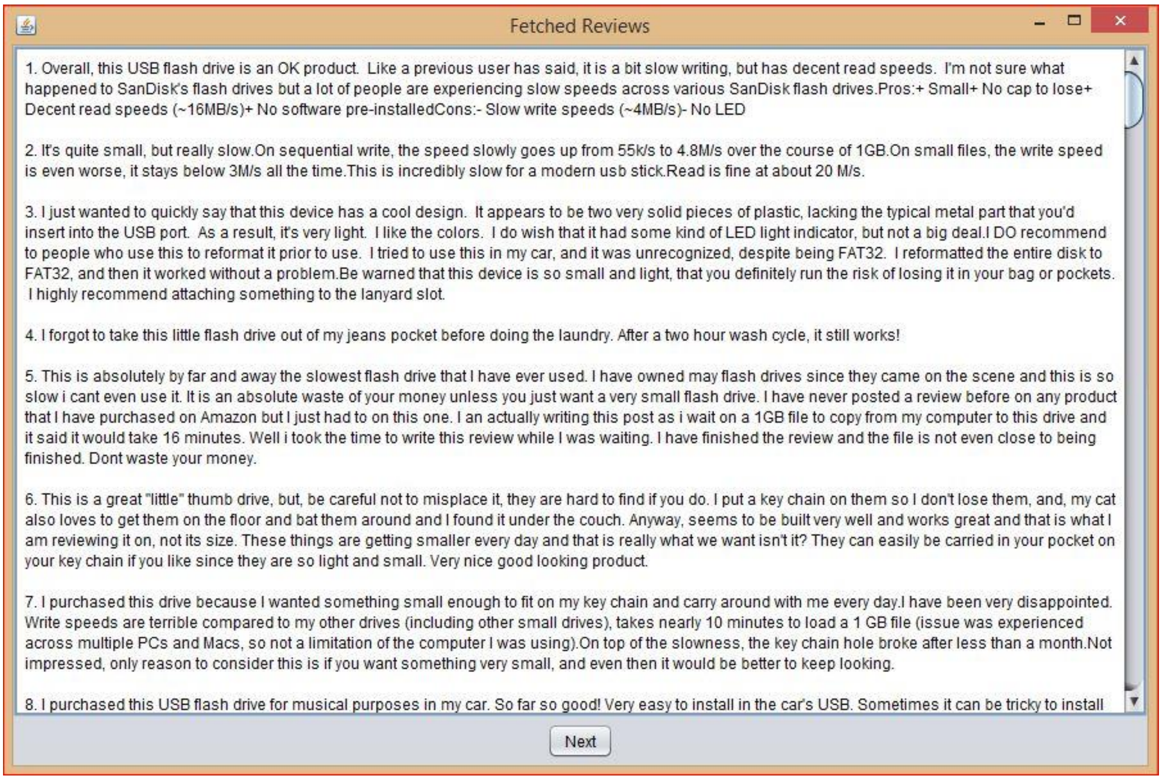


Fig. 8. Window of fetched Review

Popup in Fig. 7. will come up only if the source of review selected was Amazon [28] but Fig. 8. contains the list of reviews which will come up irrespective of whether the source was internet or file. It shows all the reviews fetched from the provided source.

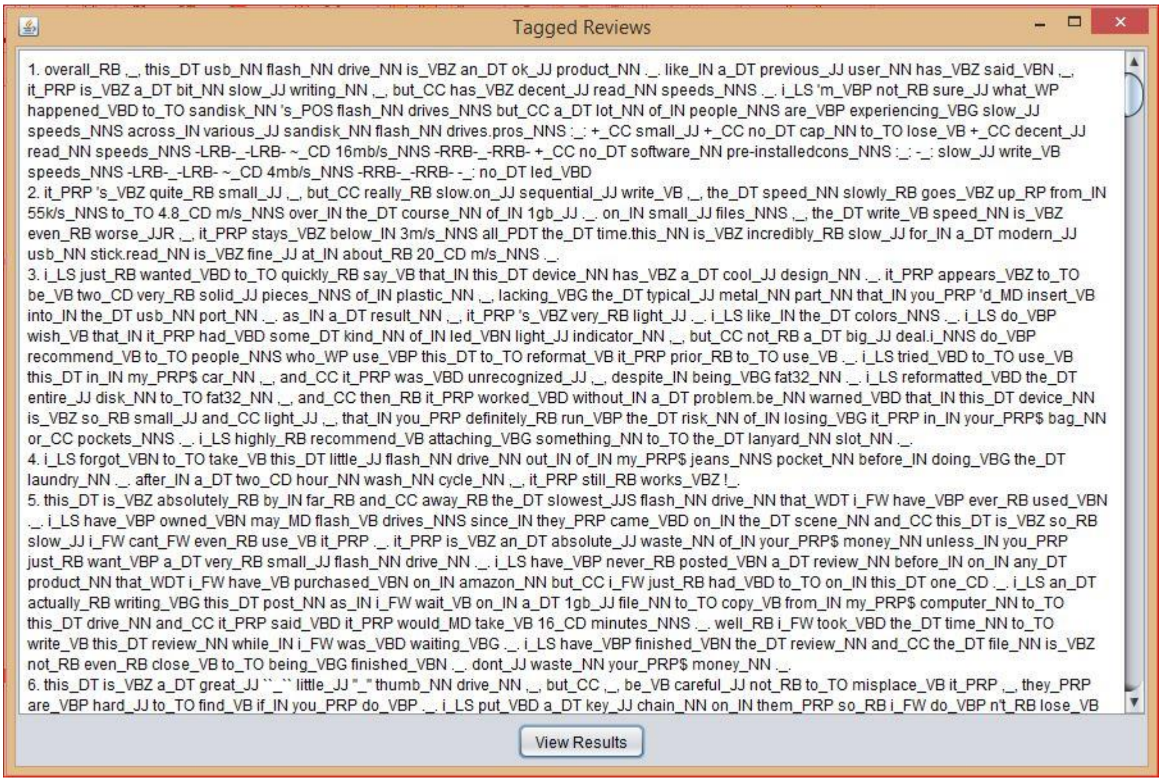


Fig 9. Window with Tagged Reviews

Fig. 9 shows the window which contains all the tagged reviews. Reviews are tagged using Maxent Taggers [30] [31] and showed as *word_tag* pair. Alphabetical list of tags with their meanings is as follows:

1. CC Coordinating conjunction
2. CD Cardinal number
3. DT Determiner
4. EX Existential there
5. FW Foreign word
6. IN Preposition or subordinating conjunction
7. JJ Adjective
8. JJR Adjective, comparative

9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	to
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive Wh-pronoun
36.	WRB	Wh-adverb

This is a list of standard tags used by maximum POS tagging systems. This list is mentioned in Penn Treebank Project [33].

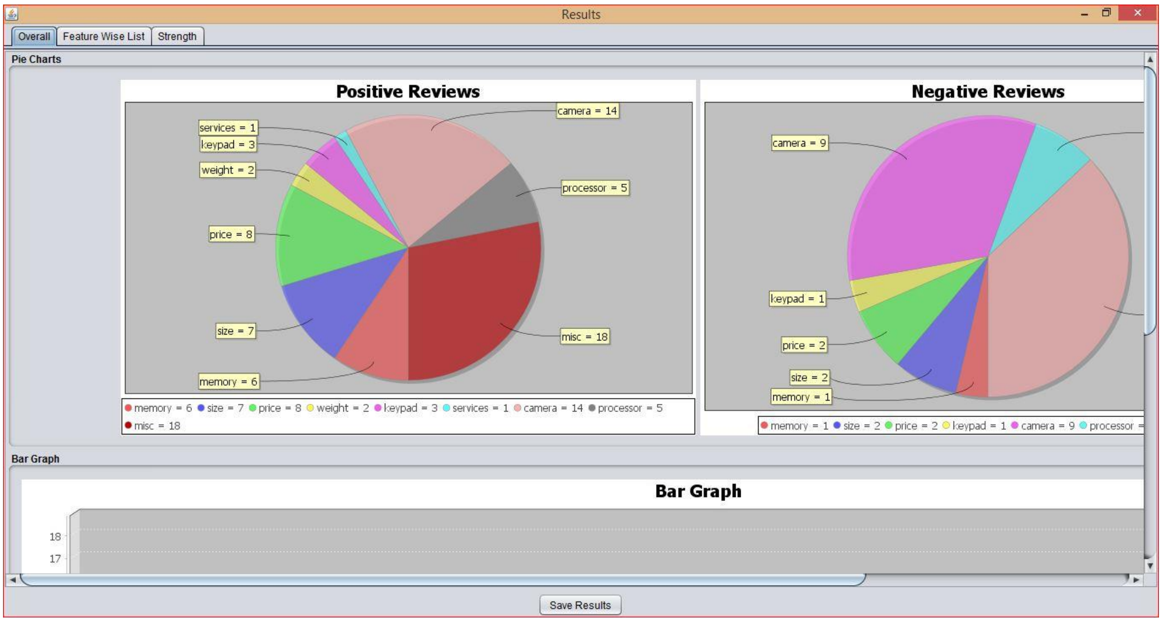


Fig. 10. Result Window (a)

Result window has 3 tabs which shows 3 different abstraction level of results. First of all it has Overall results, which contains Pie Charts and a Bar Graph showing the no. of positive and negative reviews for each feature separately. This charts are prepared using JFreeChart **[32]**.

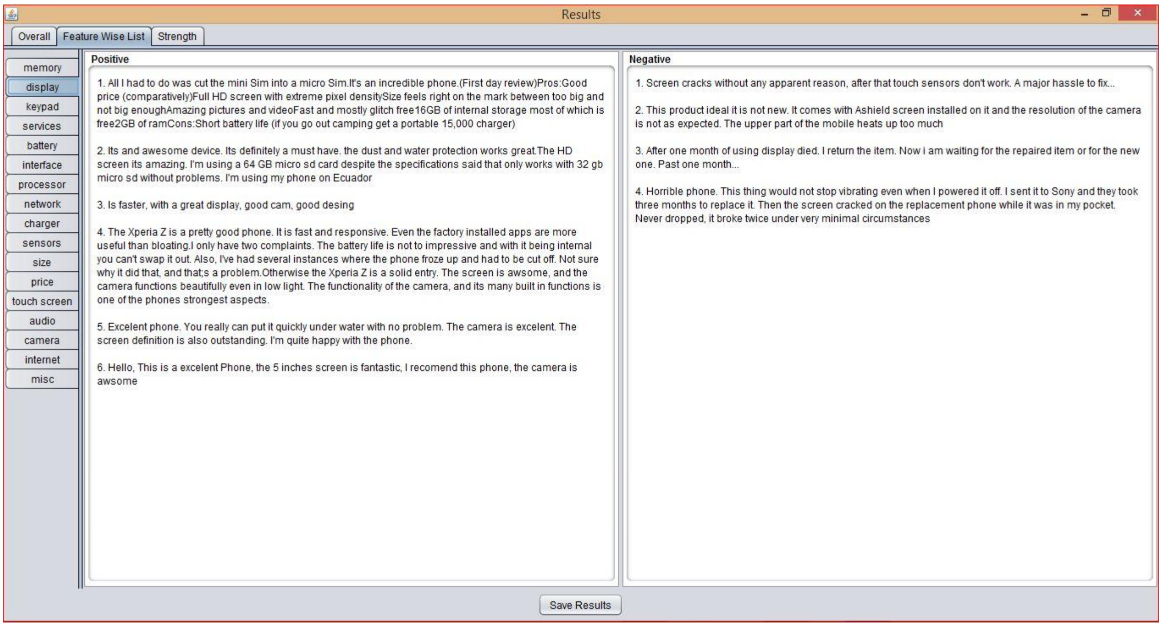


Fig 11. Result Window (b)

This window in Fig. 11 shows the list of positive and negative reviews separately and user can select a feature to view reviews related to it.

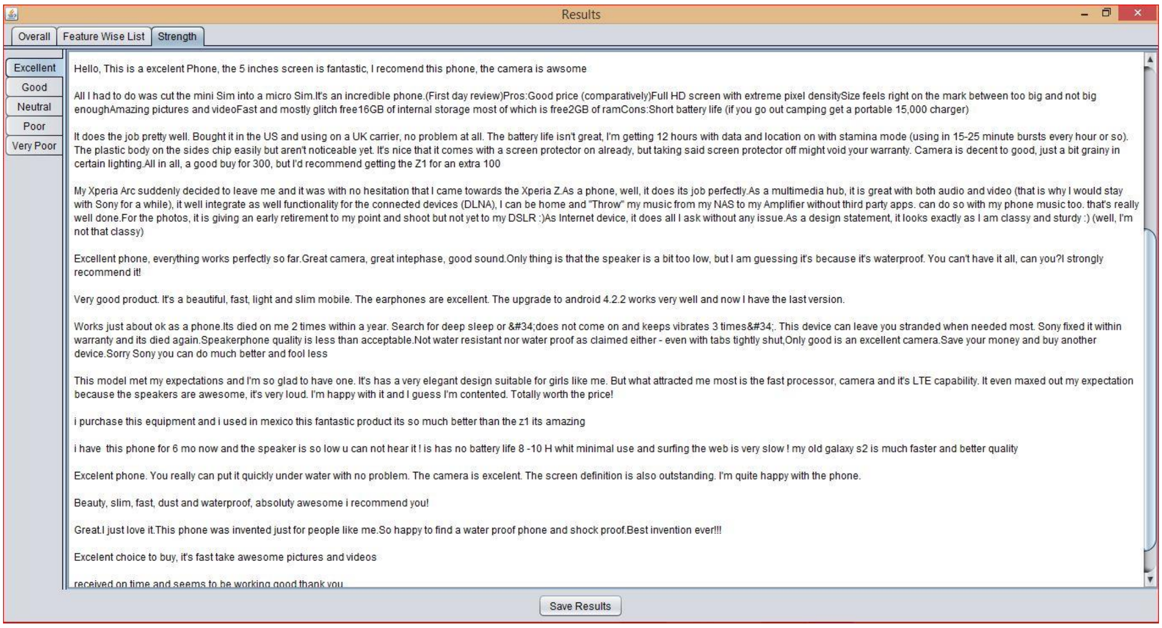
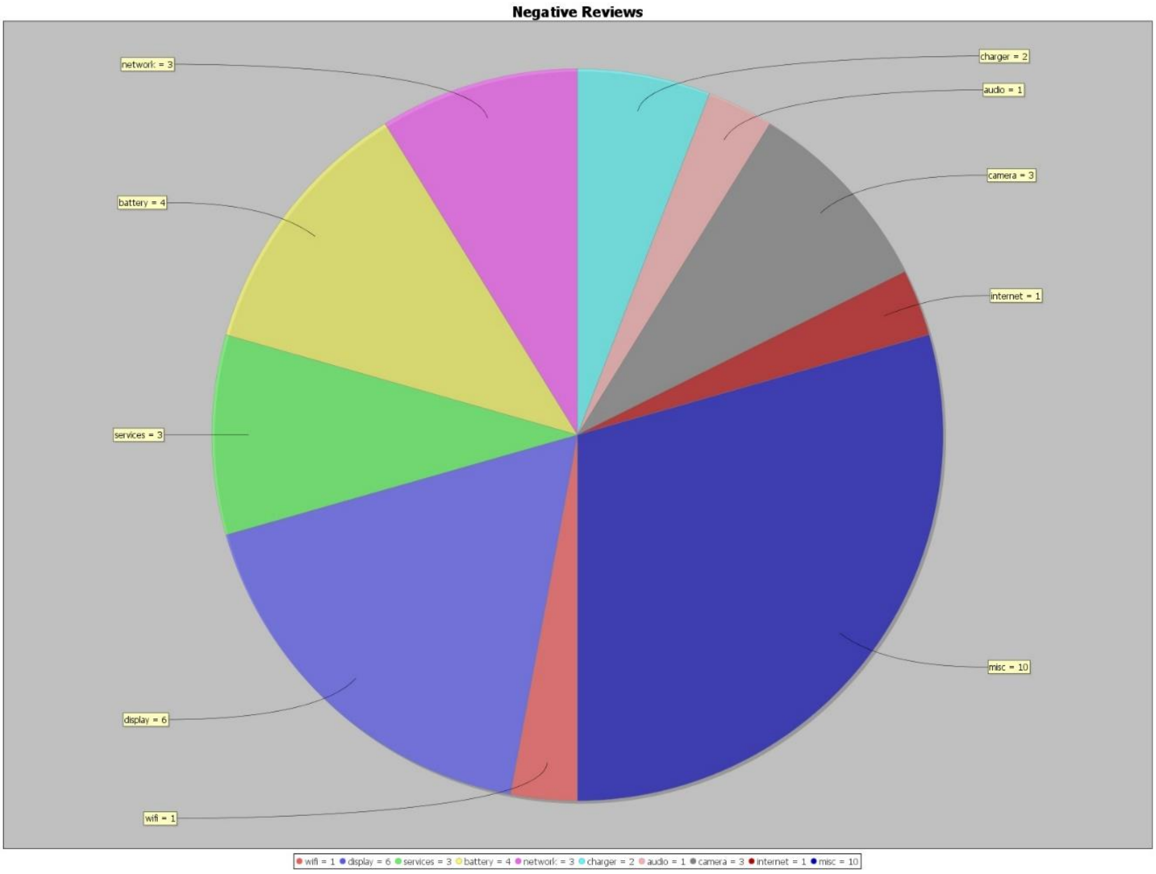


Fig. 12. Result Window (c)

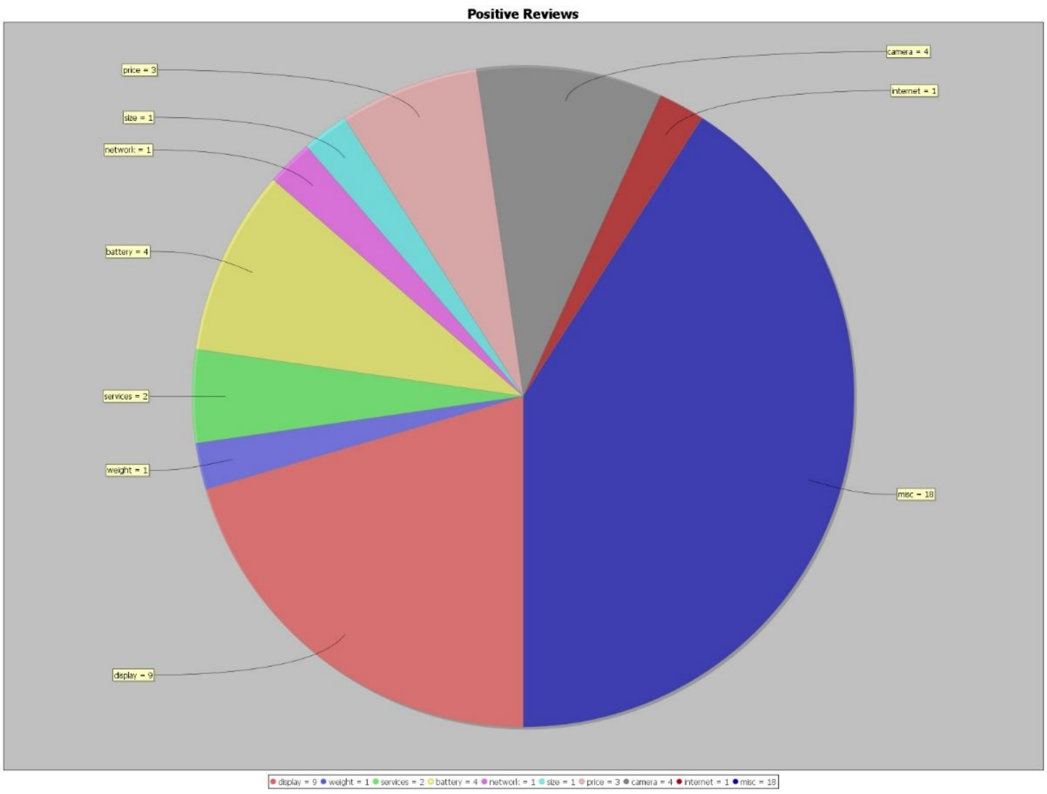
This figure shows the strength of each review. Basically the reviews are sorted according to strength of their emotions. System classifies all the reviews in 5 different classes as follows:

1. **Excellent:** Describes strong positive emotions.
2. **Good:** Describes normal positive emotion.
3. **Neutral:** Either describes equal no. of positive and negative emotion or describes no emotion toward any of the feature or the product.
4. **Poor:** Describes normal negative emotion.
5. **Very Poor:** Describes strong negative emotion.

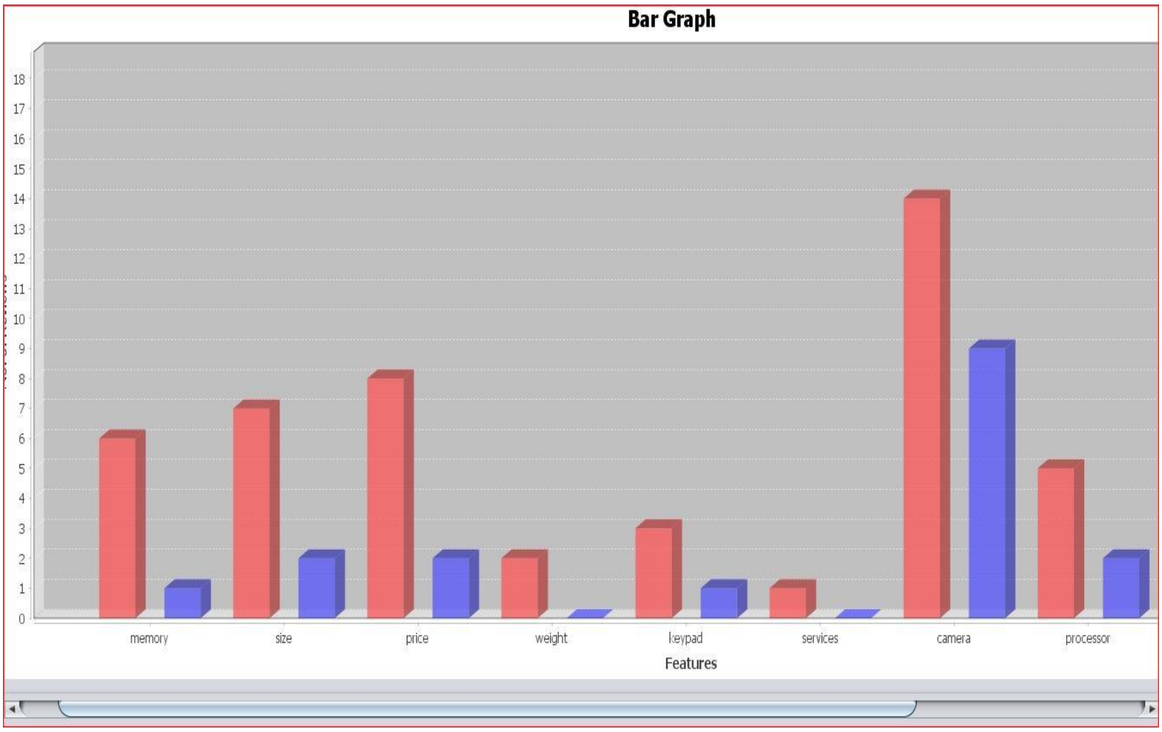
This level of result gives a crystal clear idea about the product. By looking at the no. of reviews in each category a user can make his/her mind about whether to buy the product or not.


Negative Reviews
network = 3
charger = 2
audio = 1
battery = 4
camera = 3
internet = 1
services = 3
misc = 10
display = 6
wifi = 1
wifi = 1 display = 6 services = 3 battery = 4 network = 3 charger = 2 audio = 1 camera = 3 internet = 1 misc = 10

(a).


Positive Reviews
camera = 4
price = 3
internet = 1
size = 1
network = 1
battery = 4
services = 2
weight = 1
misc = 18
display = 9
display = 9 weight = 1 services = 2 battery = 4 network = 1 size = 1 price = 3 camera = 4 internet = 1 misc = 18

(b).



(c).

Fig. 13. Saved Results.

In Fig.10, Fig. 11 and Fig. 12 there is a *save result button* at the bottom of result window. Basically system lets the user to save the pie charts and bar graph in image format using that button. Fig. 13. Shows the image files saved by the system. These files are saved in a directory of user's choice.

Chapter 7

Conclusion

Numerous techniques for summarizing product reviews using natural language processing methods are discussed in this report but this work is different from those techniques. System proposed in this report allows a user to summarize a huge database of reviews and provide a graphical summary so that the user can easily interpret the quality of a given product. Results in table 1 shows that the system works pretty decently and better than previous systems which are similar to this one. This system will help manufacturers also to check the status of their product in market. For future work this system lefts, problem to deal with comparative sentences and sarcastic sentences and also to handle implicit reviews.

References

1. Liu, B., Hu, M. and Cheng, J. (2005) Opinion Observer: Analysing and Comparing Opinions on the Web. *Proceedings of International World Wide Web Conference (WWW'05)*.

2. Hu, M and Liu, B. (2004) Mining and Summarizing Customer Reviews. *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*.

3. Ding, X., Liu, B. and Yu, P. (2008) A Holistic Lexicon-Based Approach to Opinion Mining. *Proceedings of the first ACM International Conference on Web search and Data Mining (WSDM'08)*.

4. Carenini, G., Ng, R. and Zwart, E. (2005) Extracting Knowledge from Evaluative Text. *Proceedings of the Third International Conference on Knowledge Capture (K-CAP'05).*

5. Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011, June). Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* (pp. 151-160). Association for Computational Linguistics.

6. Chakrabarti, D., & Punera, K. (2011, July). Event Summarization Using Tweets. *In ICWSM.*

7. Tong, R., (2001). An Operational System for Detecting and Tracking Opinions in on-line discussion. *SIGIR 2001 Workshop on Operational Text Classification.*

8. Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification Using Machine Learning Techniques. *In Proc. of EMNLP 2002*

9. DeJong, G. (1982). An Overview of the FRUMP System. *Strategies for Natural Language Parsing. 149-176.*

10. Tait, J. (1983). Automatic Summarizing of English Texts. *Ph.D. Dissertation, University of Cambridge.*

11. Boguraev, B., and Kennedy, C. (1997). Salience-Based Content Characterization of Text Documents. *In Proc. of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization.*

12. Mani, I., and Bloedorn, E. (1997). Multi-document Summarization by Graph Search and Matching. *AAAI'97*.

13. Hearst, M, (1992). Direction-based Text Interpretation as an Information Access Refinement. In Paul Jacobs, editor, Text-Based Intelligent Systems. *Lawrence Erlbaum Associates.*

14. Sack, W. (1994). On the Computation of Point of View. *AAAI'94, Student abstract.*

15. Das, S. and Chen, M. (2001). Yahoo! for Amazon: Extracting market sentiment from stock message boards. *APFA'01.*

16. Huettner, A. and Subasic, P. (2000). Fuzzy Typing for Document Management. *In ACL'00 Companion Volume: Tutorial Abstracts and Demonstration Notes.*

17. Bruce, R., and Wiebe, J. (2000). Recognizing Subjectivity: A Case Study of Manual Tagging. Natural Language Engineering.

18. Wiebe, J., Bruce, R., and O'Hara, T. (1999). Development and Use of a Gold Standard Data Set for Subjectivity Classifications. *In Proc. of ACL'99.*

19. Popescu, A. M., Etzioni, O. (2005). Extracting Product Features and Opinions from Reviews, *In Proc. Conf. Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, British Columbia, 339–346.*

20. Abulaish, M., Doja, M. N., & Ahmad, T. (2009). Feature and opinion mining for customer review summarization. *In Pattern Recognition and Machine Intelligence (pp. 219-224). Springer Berlin Heidelberg.*

21. Parikh, R., & Movassate, M. (2009). Sentiment analysis of user-generated twitter updates using various classification techniques. *CS224N Final Report, 1-18.*

22. Barbosa, L., & Feng, J. (2010, August). Robust sentiment detection on twitter from biased and noisy data. *In Proceedings of the 23rd International Conference on Computational Linguistics: Posters (pp. 36-44). Association for Computational Linguistics.*

23. Davidov, D., Tsur, O., & Rappoport, A. (2010, August). Enhanced sentiment learning using twitter hashtags and smileys. *In Proceedings of the 23rd International Conference on Computational Linguistics: Posters (pp. 241-249). Association for Computational Linguistics.*

24. Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford, 1-12.*

25. Dave, K., Lawrence, S., & Pennock, D. M. (2003, May). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *In Proceedings of the 12th international conference on World Wide Web (pp. 519-528). ACM.*

26. Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, *5*(1), 1-167.

27. Cambria, Erik, Bjorn Schuller, Yunqing Xia, and Catherine Havasi. (2013) "New avenues in opinion mining and sentiment analysis." *IEEE Intelligent Systems* 28, no. 2: 15-21.

28. www.amazon.com/

29. http://www.amazon.com/gp/seller/asin-upc-isbn-info.html

30. Kristina Toutanova and Christopher D. Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pp. 63-70.

31. Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of HLT-NAACL 2003*, pp. 252-259.

32. www.jfree.org/**jfreechart**/

33. Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, *19*(2), 313-330.

34. http://www-nlp.stanford.edu/

Appendix

List of Abbreviations:

AI: Artificial Intelligence

API: Application Programming Interface

CRF: Conditional Random Fields

FRUMP: Fast Reading Understanding and Memory Program

HMM: Hidden Markov Model

IMDB: Internet Movies Database

NLP: Natural Language Processing

POS: Part of Speech

SVM: Support Vector Machine

Publication