COMPARATIVE TEXT SUMMARIZATION OF PRODUCT REVIEWS

by

DINESH REDDY SINGI REDDY

B.E., Osmania University, 2008

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2010

Approved by:

Major Professor
Dr. William H. Hsu

Abstract

This thesis presents an approach towards summarizing product reviews using comparative sentences by sentiment analysis. Specifically, we consider the problem of extracting and scoring features from natural language text for qualitative reviews in a particular domain. When shopping for a product, customers do not find sufficient time to learn about all products on the market. Similarly, manufacturers do not have proper written sources from which to learn about customer opinions. The only available techniques involve gathering customer opinions, often in text form, from e-commerce and social networking web sites and analyzing them, which is a costly and time-consuming process.

In this work I address these issues by applying sentiment analysis, an automated method of finding the opinion stated by an author about some entity in a text document. Here I first gather information about smart phones from many e-commerce web sites. I then present a method to differentiate comparative sentences from normal sentences, form feature sets for each domain, and assign a numerical score to each feature of a product and a weight coefficient obtained by statistical machine learning, to be used as a weight for that feature in ranking various products by linear combinations of their weighted feature scores. In this thesis I also explain what role comparative sentences play in summarizing the product. In order to find the polarity of each feature a statistical algorithm is defined using a small-to-medium sized data set. Then I present my experimental environment and results, and conclude with a review of claims and hypotheses stated at the outset. The approach specified in this thesis is evaluated using manual annotated trained data and also using data from domain experts. I also demonstrate empirically how different algorithms on this summarization can be derived from the technique provided by an annotator. Finally, I review diversified options for customers such as providing alternate products for each feature, top features of a product, and overall rankings for products.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my major professor, Dr. William Hsu, for introducing me to the fundamentals of machine learning. I am grateful to him for guiding me in writing this thesis and helping me by making corrections and revisions to this manuscript.

I am also thankful to Dr. Gurdip Singh and Dr. Torben Amtoft for their help and guidance throughout the construction of this work and for generously offering their time and expertise to better my work.

Dedication

I dedicate this thesis to the most important people in my life
My father Kesava Reddy, mother Satyavathi and sister Divija

Chapter 1 - Introduction and Background

1.1 Introduction

Determining what others think about some entity of interest is an important piece of information for most users during the decision making process (Pang & Lillian, 2008). The last decade of the 20th century, when the World Wide Web was much smaller in scope and used less as a decision support reference than it is today, users tended to ask their friends or neighbors for suitable or alternate product recommendations. However, Kondrak (2008) writes that "with the ever-growing popularity of online media such as blogs and social networking sites the internet has became a valuable source of information for product reviews". This development has been bolstered by marked growth in public awareness of the World Wide Web later in the decade. Users can get a great deal of critical information about products from people whom they never met, but most of them do not have time to read reviews from all of these users and are unaware of all the alternate products in the market. To solve this problem researchers started working on dealing with the computational treatment of opinion, sentiment and subjectivity in text. The task of determining the attitude of a speaker with respect to various topics is known as *sentiment analysis*.

Sentiment analysis is not only useful for customers, but also helps companies to analyze opinions and attitudes of customers towards their company and its products, *i.e.*, the companies can get feedback about its products directly from customers from social networking sites such as Twitter and blogs. In this way market intelligence is created. "Information can be useful only when it is transferred to knowledge." (Shabrang) By applying sentiment analysis as a step withing text mining, unstructured data from online text is transformed into structured information.

Many sentiment algorithms were written either to determine whether input sentences in a natural language are subjective or objective, or whether they are positive or negative. The first paper on such an end-to-end sentiment analysis system was published (Hatzivassiloglou & McKeown, 1997). Many researchers worked on this task and proposed different algorithms for different conditions. These algorithms were written based on real-life needs such as movie reviews, which lead to a commercial success of this system, and most of the leading companies in product manufacturing depending upon this sentiment analysis algorithm. Companies such as

Sentimetrix, Jane16, sensenet etc… are providing regular trends to their clients on daily or monthly basis. Despite of lot of research in this area, there are lot of loop holes and arguments for algorithms of domain specific, and in deciding the polarity in different types of sentences.

Especially in deciding the polarity of whole document, an argument regarding giving equal importance to all features of a product comes into scene. This thesis paper gives a algorithm for deciding the polarity of each product from the data of all documents, and gives the score to each feature and that score depends on other product that is compared to this. Here I am giving different weight age to each feature of the product and rate the product or give its polarity and when there is a comparative sentence on this product then there is an adjustment to the rating of both the products in the comparative sentence based on the polarity of that sentence.

1.2 Background

Presently a lot of research work is going on in this field of sentiment analysis. Many popular algorithms have been used for sentiment analysis. For all the algorithms used may be document level or feature level the common term used is sentiment analysis feature which is a measurable property of a document or sentence ready for sentiment analysis such as polarity or frequency. A Sentiment Analysis algorithm depends on this sentiment analysis feature. There can be many algorithms that use one feature and one algorithm that used many features (Esuli & Sebastiani, Determining the semantic orientation of terms through gloss classification, 2005) (Esuli & F, SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining, 2006) (Esuli & Sebastiani, Pageranking wordnet synsets: An Application to Opinion Mining, 2007). Other terms we use are users are those who post reviews in websites or express their opinion in social network websites. Pang and Lee and many other researchers conducted various researches on designing new algorithms and all sentiment analysis algorithms can be classified into two types sentiment lexicon based algorithms (Esuli & Sebastiani, Determining the semantic orientation of terms through gloss classification, 2005) (Turney P. D., 2002) and machine learning based approaches (Pang, Lillian, & Shivakumar, 2002). All terms in the dictionary can be labeled by sentiment lexicon with polarity information using positive and negative scores. Few features can be obtained through this sentiment lexicon (Zhe, 2010).

There are two different approaches in sentiment analysis, one is statistical technique and the other is linguistic technique. Statistical technique is a mathematical technique where opinion depends on the number of positive and negative statements in the text, where as linguistic technique is build on a set of rules and compares the text to be analyzed with them (Wikipedia, 2009)

1.3 Motivation

A large amount of data in the form of reviews on various smart phones is available online in unstructured format. To exploit this unstructured data sentiment analysis techniques and Natural Language Techniques can be used to decide the polarity.

Consider the following example

Ex: : I hate the phone. It's the worst one I have used

After using Sentiment analysis program and Parts of speech tagging

<SENTIMENT>

<ENTRY certainty="100.0" positiveHits="0" negativeHits="2" positivePolarity="0.0" negativePolarity="434.18">NEGATIVE</ENTRY>

</SENTIMENT>

- The negative polarity and positive polarity is the number of times, the phrase in the sentences found in database.

The polarity of a single sentence is shown, to determine the polarity of whole document and to determine the total polarity score of a product; we need to summarize all the sentences, It's been done many times by many algorithms but all algorithms gave equal weight age to all the features and made decision based on the statistics such as, if total number of positives features is less than negative features then the product is given positive review, but not all features in a smart phone or in any product have same value. Duplicate reviews which may affect the decision making algorithm. All these things are not considered in a single algorithm while making decisions.

1.4 Problem Statement

Suppose for n products(smart phones) the number of documents be d_n. All the documents are from the same source C and of same in content (equal number of reviews). First all documents are chunked into sentences and then we classify comparative sentence and normal sentence. A comparative sentence is one which has at least two smart phone names. The tasks will be in the following way.

Task1: A comparative sentence is obtained using smart phone extractor, which is created using a set of smart phone names. If smart phone extractor finds two different smart phone names along with a comparative word in a single sentence it decides that corresponding sentence as comparative sentence.

Task 2: If smart phone extractor finds only one product then it is considered as normal sentence and polarity of each feature is identified.

The polarity of each feature is identified. For two taggers were created using the sets defined for corresponding purpose. In addition to these we need to eliminate the following things.

i) Duplicate reviews from the data set.
ii) Multiple reviews from the same author.
iii) Sentences such as "EVO phone is a contender against the I-Phone 4." This one is a comparative sentence as per smart phone extractor but do not have any feature.

The structured information that we are extracting is as shown below:

i) Smart phone names ("I-Phone", "Evo", "Android", "Droid X " etc…)
ii) Comparative words identification.
iii) Comparative Vs Non comparative
iv) Features and Phrases ("music" "music player" etc…)
v) Extract Information with respect to feature and also with respect to phone.
vi) Summarize the reviews automatically by considering the comparative sentences that we extracted using comparative words and smart phone names.

1.5 Goal

Goal: Providing complete feature level polarity information to customers for each product and also providing comparative score between two products of same feature.

In this thesis I concentrated on using all features in deciding the polarity of whole document, I had given individual weightage to each feature so that each feature has very restricted role in deciding the polarity of whole product in the document. In this thesis instead of just summarizing the reviews, I considered of using scoring for each feature of the product and also introduced negative scoring for negative polarity in a comparative sentence. The below example shows why it is important to introduce negative scoring in comparative sentence.

Ex: the display of I-Phone is better than that of Evo.

In this comparative sentence the I-Phone has positive polarity and Evo has negative polarity, in regular sentiment analysis problem, I-Phone gets positive score, this positive score gives advantage to I-Phone but it should give a negative score to Evo, then we can say that this users view on Evo is also counted. In this thesis I had given an equal amount of negative score to the Evo, because I-Phone's advantage should be a loss to its competitor.

There is no need to use association mining to filter features and phrases as did in Mining and summarizing customer reviews by Bing Liu and Minging Hu (Hu & Liu, 2004) as separate sets were introduced to filter the features. It is important to use separate sets of Smart phones and features and sets for scoring each feature because it removes all unimportant data from the result. Sets defined in this thesis include

1. sets for identifying the phones.
2. set for identifying the features of products. [previously POS tags such as /NN is used for identifying the feature and then researchers used to reduce them by association mining]
3. 3 sets for scoring each feature in deciding the polarity of whole product

Finally we build a system where users can view products based on features and customers can find the polarity of each feature of the product and their corresponding scores. Here in this thesis I use both Statistical and linguistic techniques.

The technical significance of this thesis project is that it shows how to do opinion mining in comparative sentences in document level. Till now comparative sentences are mined in

sentence level *i.e.,* algorithms are defined to determine the polarity of a comparative word in a sentence, here we use the same strategy but by considering each document as set of multiple sentences and each sentence is mined separately and all sentences are summarized collectively by considering all features.

Throughout this thesis, the term "feature" generally refers to a object feature. For example: "An opinion passage on a feature f of an object O evaluated in d is a group of consecutive sentences in d that expresses a positive or negative opinion f" (Liu, 2010).

1. Prove that this algorithm when compared to other algorithm from Bing Liu performs reasonably well when feature sets are considered and scoring, comparative sentences are not considered.
2. Diversify the options that the user will have for each product and show other alternatives of each product based on the feature that customer choose without major deviation in polarity of the dataset.

Finally this thesis provides a complete summarization of reviews of a product along with a recommendation system on product based on features. The major contribution of this thesis is one can see how many diversified options can a sentiment analysis system generate including that of recommendation system and this detailed analysis was never done before.

1.6 Outline

The rest of the thesis is organized as follows

Chapter 2: In this chapter a brief discussion of previous sentiment analysis algorithms and techniques are discussed. I also give an overview of few web resources that summarizes various trends.

Chapter 3: We give a overall frame work for design decisions which include rationale and alternatives, scope of the work. The background information of few techniques and tools which are used is described. The purpose of using those techniques and tools is also explained. Creating Feature sets for scoring and explained the reasons to create each set. The methodologies used in this thesis are discussed in this section along with the novel contribution i.e., a complete system architecture is discussed in this section.

Chapter 4: This chapter mainly deals with test bed and framing problem. Explained how the corpus is developed or collected. The experimental setup along with evaluation criteria are discussed in this section. Step by step result of preliminary experiments [Experiments done with small amount of data] is shown and explained the progress of the experiment in detail.

Chapter 5: The results for the experimental setup done in chapter 5 are shown in this section. Interpretation of results is done by comparing with other algorithms and shown how users are given multiple options for each product that they choose.

Chapter 6: We conclude with limitations of approach, a list of contributions, and future work proposals.

Chapter 7: Gives you the complete bibliography used for this thesis.

Chapter 2 - Related Research

2.1 Sentiment Analysis

A sentiment is a thought or an attitude or a view, which is based mainly on emotion but not on reason. Sentiment Analysis is a natural language processing and computational techniques to automate the extraction or classification of sentiment from typically unstructured text. Opinions are important because whenever we need to make a decision we listen to others. Opinions are found at various levels document level, sentence level and feature level. In this project we are dealing at basic feature level. Whenever a sentence is given to identify its opinion the sentiment analysis system looks for the following template.

<opinions, target of opinions and opinion holders>

2.1.1 Document Level Sentiment Classification

Here we find the overall opinion of the document. The document is classified based on overall sentiment expressed by single opinion holder. Here the document must focus on single object. (Pang & Lillian, 2008)

Classes : positive and negative

2.1.2 Sentence Level Sentiment Analysis

For sentence level sentiment analysis (Wilson, Wiebe, & P, 2005) will have two tasks:

i) Subjectivity classification

ii) Sentiment classification: this is for subjective sentences gives polarity i.e., classifies as positive or negative.

Subjective classification classifies sentence as subjective or objective

Objective: I brought a camera.

Subjective: it was a nice camera.

This subjective example can be classified by sentiment classifier as *positive.*

2.1.3 Feature Based Sentiment Analysis

Feature based sentiment analysis gives a much detailed information about a product. Through document and sentence level sentiment analysis if a product X is good, it doesn't mean that all features of that product are good. So a feature based sentiment analysis is introduced to identify the polarity of each feature. The term feature represents both component and attribute.

2.1.4 Opinion Types

There are two main types of opinions they are

i) Direct Opinion: Direct sentiment expressions on any target objects.
   Example: The I-Phone 4 has an excellent display.
ii) Comparative Opinion: "Comparisons expressing similarities or differences of more than one object. Usually standing an ordering or preference." (Pang & Lillian, 2008)
   Example: I-Phone 4 has larger battery life than evo.

2.2 Algorithms for Sentiment Analysis

As mentioned in the previous chapter, there are two important types of algorithms or most of the algorithms are of either sentiment lexicon based or machine learning based. Both types of algorithms can be used for determining subjective or objective sentences and also in determining in positive or negative sentences.

2.2.1 Sentiment Lexicon Based Algorithms

Most of the unsupervised sentiment analysis algorithms fall into this category. This is more a statistical technique. The algorithms of this type construct functions to calculate the polarity of review of feature. They calculate the positive scores and negative scores of each feature and calculate the overall positive and negative degree of each review and also calculate the average term positive score and average term negative score, if average term positive score is more than that of average term negative score then it is considered as positive in sentiment.

The basic step for this type of algorithms is to do sentiment orientation for the lexicon words. This can be done in two ways, either by Esuli way of using word net or by using turner's

mutual information and co-occurrences. Esuil's SentiWordNet is a lexical resource in which for each WordNet synset(s) there are three numerical values they are Pos(s), Neg(s) and Obj(s), explaining how positive negative and objective the words containing in synset are. The three scores are derived by a combining the results of ternary classifiers (Esuli & Sebastiani, 2005). The main method used to develop SentiWordNet is quantitative analysis of the glosses associated to synsets and on the use of semi-supervised synset classification.

The figure below shows the graphical representation adopted by SentiWordNet for representing the opinion related properties of a term sense.

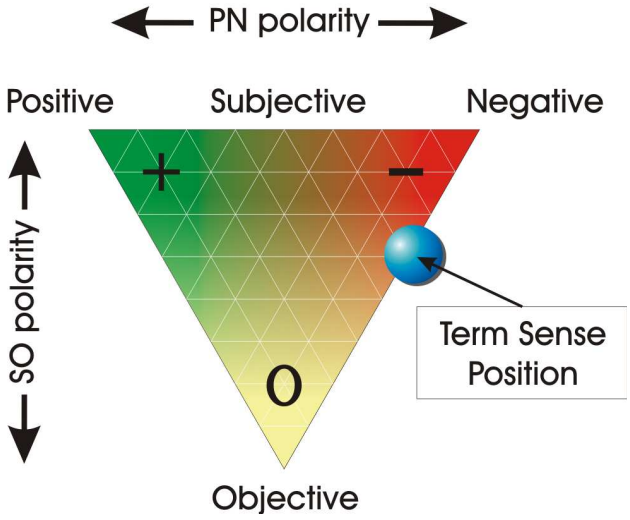
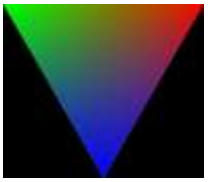


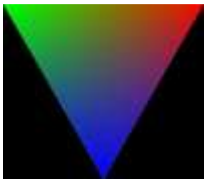
Figure 2.1 SentiWordNet general visualization

The below images are few screenshots of the output for the term *good*

- Adjective

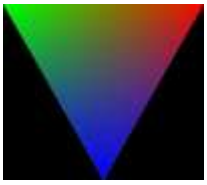


P: 0.875 O: 0.125 N: 0



P: 0 O: 1 N: 0

- Noun

 P: 0.5 O: 0.5 N: 0

Coming to Turney method of mutual information and co-occurrence, the problem of determining the orientation of terms is approached by conducting the below three steps:

i) Tag the data to identify parts of speech and phrases. (phrases contain either adjectives or adverbs)

ii) Find the polarity of each phrase

iii) Review is recommended or not recommended based on the average semantic orientation score of whole review.

Starting from the first algorithm by Hatzivassiloglou (Hatzivassiloglou & McKeown, 1997) to the present dates most of the algorithms deals with adjectives and even in this the phrase contains adjectives because adjectives indicate subjective sentences. In this thesis I use phrases instead of words because it helps in identifying the context, so that semantic orientation can be improved to a great extent.

Example:

Consider the word *"Unpredictable"* may have a negative orientation in one domain and positive orientation in other domain such as

"Unpredictable steering" has a negative orientation in automotive review

"Unpredictable plot" has a positive orientation in movie review. (Turney, 2002)

So if we use phrase as per Turney, even if one word is adjective the other word provides the context. Turney defined few pattern of tags for extracting two word phrases from a reviews such as:

i) First word is an adjective, second word is noun, third can be anything.

ii) First word is an adverb, second word is an adjective, third word cannot be noun.

In the similar way he defined 5 patterns to extract phrases and then estimated the semantic orientation of the extracted phrase. For this PMI (pointwise mutual information) between two words is:

$$PMI(word_1, word_2) = log_2\left(\frac{p(word_1 \,\&\, word_2)}{p(word_1)p(word_2)}\right)$$

Semantic orientation of phrase is calculated as

SO(phrase) = PMI(phrase, "excellent") - PMI(phrase, "poor")

PMI-IR estimates PMI by giving queries to search engine and noting the number of hits.

$$SO(phrase) = log_2 \frac{hits(phrase\ NEAR\ \text{"excellent"})hits(\text{"}poor\text{"})}{hits(phrase\ NEAR\ \text{"poor"})hits(\text{"excellent"})}$$

Apart from these two there is also another algorithm using Page ranking (Esu07).

There are few researchers such as Kamps, Marx (Kamps, Marx, & Rijke, 2004) who focused for using the lexical relationship defined in WordNet. They draw a graph on adjectives which are at intersection between TL term set and WN, they add a link between two adjectives if WN indicates a synonymy relation between two.

$d(t_1,t_2)$ is the shortest path that connects t_1, t_2 terms.

If $d(t_1,t_2)$ = + infinity then t_1 and t_2 are not connected.

The orientation of term is determined by its relative distance from seed terms *good* and *bad.*

$$SO(t) = \frac{d(t, bad) - d(t, good)}{d(good, bad)}$$

This approach is very limited one, because there will be very few adjectives (663 as per author) reachable from either good or bad seed terms.

As an extension to the Turney work on mutual information; Turney and Littman (Turney & Littman, 2003) approached the problem of determining the orientation of terms by bootstrapping from two seed sets.

The seed sets defined by them are

S_p = {good, nice, excellent, positive, fortunate, correct, superior}

S_n = {bad, nasty, poor, negative, unfortunate, wrong, inferior}

Even this method is based on the Pointwise mutual information. For a given term *t* the orientation value O(t) is given as:

$$O(t) = \sum_{t_i<s_p} PMI(t, t_i) - \sum_{t_i<s_n} PMI(t, t_i)$$

2.2.2 Machine Learning Based

Machine Learning based algorithms are supervised; the regular techniques in this type are Support Vector Machines, Maximum Entropy, and Naïve Bayes. All the machine Learning based methods are compared in the research work by Pang and Shivakumar (Pang, Lillian, & Shivakumar, 2002). The three Machine learning methods are discussed and compared below.

2.2.2.1 Naïve Bayes

Assumptions for all the 3 methods:

Let $\{f_1, f_2, \ldots, f_m\}$ be a predefined set of features in a document

Let $n_i(d)$ be the number of times f_i occurs in document d.

Then document d is represented as

$d: = (n_1(d), n_2(d), \ldots, n_m(d))$.

The naïve bayes classifier when applied to text classifier of document d, of

class $c = \text{argmax}_c P(c/d)$ is

$$P(c|d) = \frac{P(c)P(d|c)}{p(d)}$$

P(d) plays no role. Now by assuming all f_i's are conditionally independent Naïve Bayes decomposes P(d/c) to estimate it.

$$P_{NB}(c|d) = \frac{P(c)\left(\prod_{i=1}^{m} P(f_i|c)^{n_i(d)}\right)}{p(d)}$$

This method consist of frequency estimation of P(c) and P(f_i | c)

Pang concluded that despite its simplicity and its conditional independence assumption it performs well.

2.2.2.2 Maximum Entropy

Using Maximum entropy the P(c | d) value is given as:

$$P_{ME}(c|d) := \frac{1}{Z(d)} exp\left(\sum_i K_{i,c} F_{i,c}(d,c)\right)$$

Z(d) : Normalization Factor

Fi,c : Feature / Class function for feature fi and class c

$$\text{Fi,c (d, c`)} := \begin{cases} 1, n_i(d) > 0 \; and \; c` = c \\ 0 \;\; otherwise \end{cases}$$

As is the case with naïve Bayes, this maximum entropy has no assumptions.

2.2.2.3 Support Vector Machines

This is the most effective method for text categorization. The basic idea here is to find a hyperplane (w), which separates the document vectors in one class from those in the other class, having separation as large as possible.

Allowing c_j to {1, -1} the solution is

$$W := \sum_i \alpha_j \, c_j d_j, \alpha_j \geq 0$$

	Features	# of features	Frequency or presence?	NB	ME	SVM
1	Unigrams	16165	freq.	**78.7**	N/A	72.8
2	unigrams	``	pres.	81.0	80.4	**82.9**
3	unigrams + bigrams	32330	pres.	80.6	80.8	**82.7**
4	Bigrams	16165	pres.	77.3	**77.4**	77.1
5	unigrams + POS	16695	pres.	81.5	80.4	**81.9**
6	Adjectives	2633	pres.	77.0	**77.7**	75.1
7	Top 2633 unigrams	2633	pres.	80.3	81.0	**81.4**
8	unigrams + positions	22430	pres.	81.0	80.1	**81.6**

Table 2.1 Average three fold cross validation accuracies, in percent.

Boldface: Best performance for a given setting (row) (Pang, Lillian, & Shivakumar, 2002).

2.3 Bing liu Summarization Method

This project deals with summarization of reviews using comparative sentences. Bing liu suggested a summarization technique for summarizing reviews but without considering the effect of comparative sentences. (Hu & Liu, 2004)

According to bing method the summarization steps are as below.

i) Identify features of the product that customers have expressed their opinion on.

ii) For each feature we need to identify review sentences that give positive or negative opinions

iii) Produce summarized results.

The block diagram used for bing method is shown below (Fig 2.5). The Feature is something with /NN as POS tag and /NN /NN as phrase.

The method worked well with 0.69 recall and 0.64 precision, but has its own limitations such as giving summarized results of a particular feature. One cannot decide the polarity of a product by using those results because here all product are of equal important.

The other disadvantages of this method are to use POS tags for identifying feature and phrases. I eliminated this disadvantage by using a pre defined feature set for each domain and there by reduced the burden on system by eliminating non features much before finding its polarity.

At the end of this paper the authors concluded that they did not determine the strength of the opinion, The main contribution s' opinion on a particular product by giving weight age to each feature.

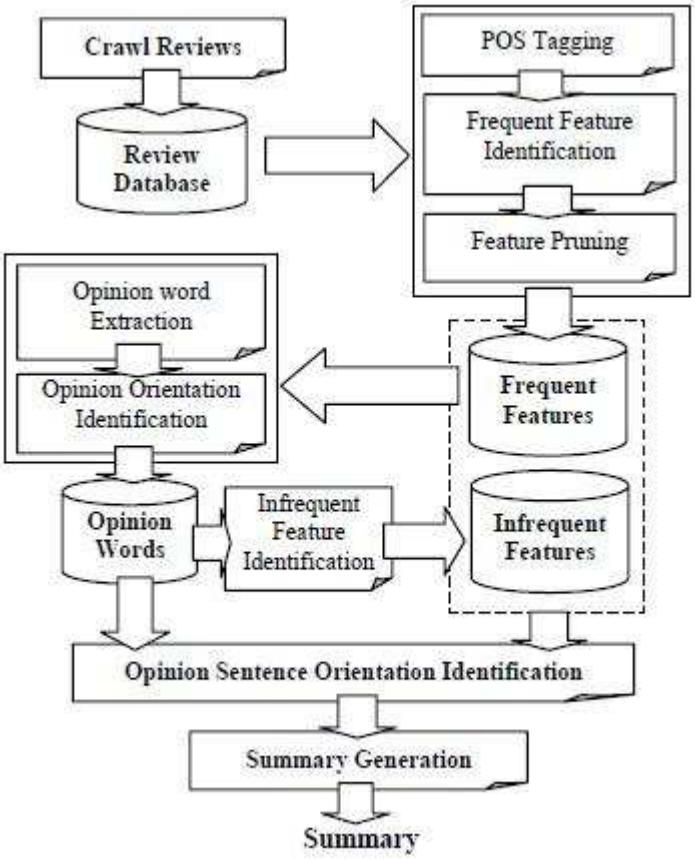


Figure 2.2 Bing Liu feature based opinion summarization

Chapter 3 - Methodology

This chapter presents my approach for comparative text summarization of product reviews by comparative sentences using sentiment analysis, which I had done in several phases starting by collecting enough data or reviews by crawling and then filtering structural information from this crawled unstructured information by using feature set. Separated comparative sentences from normal sentences, then developed algorithm and used tools to find the polarity of each feature towards that product in both normal and comparative sentence. Finally, analyzed the results to determine top feature's of the product and top product for each feature. Along with this I even discussed the evaluation technique used for this experiment.

3.1 Crawling e-commerce websites

For analyzing or for learning any new algorithm a suitable size of data is required. This data initially will be unstructured. Here I gathered part of data, using a tool web content extractor, also collected few data manually. Web content extractor (Newprosoft, 2009) is an online tool for extracting data from web, this tools enables us to extract multiple data types at a time, also by separating each one.

For example: If you are extracting information about any laptop computers from the e-commerce website, we can get data about its item (model) name, its description, its min price, max price etc. The sample output of this tool has been shown below.
In the algorithm seeds are the columns required to be selected.

Algorithm 1: Information Retrieval Functionality

Input: Set S of seeds $s_p \in S$ and set T of terms $t_i \in T$, set of topics K.
Output: collection D of documents $_{dj}$, set of documents R^q relevant to query q.
doCrawl($S; T$);
[$D \rightarrow K$] = classifyDocsByTopics(D);
i = indexDocuments(D);
if $q \in$ {*Smart phone*};
[*Rsp*] = searchBySmart phone(*sp;D*);
end;
end;

ID	Item Name	Description	Price (min)	Price (max)
26	Toshiba Tecra M4-S435 1.73 GHz Pentium ...	Windows XP Tablet PC Edition - 60 GB Har...	$1,256	$2,049
27	Dell Inspiron B120 1.4 GHz Celeron M 360 L...	Windows XP Home - 40 GB Hard Drive - 51...	$499	$690
28	Toshiba Satellite A105-S361 2.0 GHz Intel ...	120 GB Hard Drive - 1 GB RAM - 15.4 in Sc...	$1,088	$1,500
29	Toshiba Libretto U105 1.2 GHz Pentium M L...	Windows XP Pro - 60 GB Hard Drive - 512 ...	$1,125	$1,949
30	Apple MacBook Pro 1.83 GHz Core Duo Lap...	Mac OS X v10.4 Tiger - 80 GB Hard Drive -...	$1,834	$2,004
31	Toshiba Qosmio G35-AV600 1.86 GHz Intel ...	DVD-RW Drive	$2,245	$2,599
32	Sony Vaio 1.73 GHz Pentium M Laptop	Windows XP Home - 80 GB Hard Drive - 51...	$799	$1,100
33	Toshiba Satellite A55-S3063 1.5 GHz Penti...	Windows XP Home - 80 GB Hard Drive - 51...	$790	$1,150
34	Toshiba Satellite 1.73 GHz Pentium-M 740 L...	Windows XP Home - 100 GB Hard Drive - 1...	$950	$1,300
35	Lenovo ThinkPad T43 1.7 GHz Pentium M L...	Windows XP Pro - 40 GB Hard Drive - 256 ...	$1,200	$1,399
36	Apple PowerBook G4 1.5 GHz PowerPC G4 ...	Mac OS X, Mac OS X 10.3 Panther - 80 GB...	$1,394	$1,704
37	Apple PowerBook G4 1.67 GHz PowerPC G...	Mac OS X, Mac OS X v10.4 Tiger - 120 GB ...	$1,930	$2,807
38	Motion Computing LE1600 1.5 GHz Pentium...	Windows XP Tablet PC Edition - 30 GB Har...	$1,559	$2,370
39	Lenovo ThinkPad T30 1.8 GHz Pentium 4 P...	Windows XP - 40 GB Hard Drive - DVD-RO...	$479	$700
40	Acer Aspire 3623WXMi 1.5 GHz Intel Celer...	Windows XP Home - DVD-RW, DVD+RW D...	$640	$758

Figure 3.1 Crawled output using web content extractor

The output from this crawler is an unclassified and unstructured output. Apart from crawling using tools one can even collect data manually, but the main restriction for this project while crawling manually is the user who collects the data must make sure that the data he collected does not have any duplicate reviews or multiple reviews from one user. Multiple reviews or multiple reviews from one user make the results biased. So while collecting the data I made sure that I collected data by removing all spam reviews for this I used to collect reviews for a particular product only from one site instead of collecting from multiple websites. The websites such as Amazon, eBay and cnet will not allow users to post multiple reviews. If we collect reviews from multiple websites, there will be a problem *i.e.*, the review posted in one website for one product may also be posted in other websites.

3.2 Comparative Sentences Extraction

After collecting data we now focus on our main idea of using comparative sentences in summarizing about a product using its reviews. We are focusing more on comparative sentences because those provide a convincing way for evaluating a product. For example if a new product comes into market, the product owner want to the public opinion on the product, not only that he would even like to know where it stands when compared with its competitors. The only trusted way to find such information by customer reviews from e-commerce websites. In order to do this

we need comparative sentences to be filtered and should be analyzed separately from regular normal sentences.

Comparative sentences can be subjective or objective. There are 4 types of comparatives they are

i) Non-Equal Gradable: Sentences showing relations of type greater or less than *i.e,* which show some ordering of products with respect to features.

ii) Equative: If with some feature two objects are equal, such type of relations are shown are equative.

iii) Superlative

iv) Non-Gradable: sentences which compare more than one product but never grade those products.

Ex: EVO has 4g, I-PHONE has 960 by 640 pixel resolution.

Out of these 4 types of comparative sentences, we are going to deal only with the non-equal gradable sentences, because it actually compares and give some ordering between two products which solves our problem.

JJR	Adjective, comparative
JJS	Adjective, superlative
RBS	Adjective, superlative
RBR	Adjective, comparative

Table 3.1 POS tags for identifying comparative sentences

Parts-Of-Speech tags are used to identify comparative sentences. The above 4 POS tags are used to identify comparative sentences, but not all sentences with the above POS tags are comparative sentences.

Ex: “In the context of speed, faster means better.”

After POS tagging:

In/In the/DT context/NN of/IN speed/NN ,/, faster/JJR means/NNS better/JJR ./.

Here we have a comparative POS tag JJR but this is not a comparative sentence.

There are some comparative sentences without any indicator word.

3.2.1 Keywords Strategy

Here we find all the keywords that cover almost all the comparative sentences. For this keywords set I expanded nitin jindal basic keywords set. The keyword set contains all the complete list from nitin, along with this I even added few other words by finding synonyms to the list of present set using Word Net (Fellbaum, 1998). The present set from nitin consist of all *–er* words along with indicative words for comparisons, e.g., beat, outperform, exceed, etc. Also added few phrases such as *number one, up against, unbeatable, on the other hand, as far as, as long as, but, whereas*. Phrases such as "as far as", "as long as", "as fast as", *i.e.*, as <word> as can be used for identifying comparative sentences.

Along with these we can also use POS tags such as JJR, JJS, RBR, RBS as keywords. My final set K consisted of all keyword set and these four tags.

K = {JJR, JJS, RBS, RBR} U {keyword set}

POS tag JJR	Prefer	Either	Number one	Twice	defeat	peerless
POS tag JJS	recommend	outperform	Superior	up against	Favor	Outdo
POS tag RBR	one of few	behind	similar	Identical	Versus	match
POS tag RBS	first	outdistance	outsell	Vs	Thrice	Unmatched
Improve	equivalent	altogether	Alternate	Outmatch	ahead	Fraction
Least	Outdistance	Outclass	unlike	Nonpareil	advantage	Outstrip
None	Win	Near	Rival	Lead	Exceed	Top
Differ	One of few	Outwit	Alternate	Compare	Dominate	Most

Table 3.2 Sample Keywords for extracting comparative sentence from unstructured data.

Identifying these keywords takes a long time, but once it is done it can be used by anyone at any point and the list can also be expanded. If more researchers start using this keyword set, they can expand this list. This may be time consuming but cost effective because we no need to label sentences as we are not using any machine learning techniques to automate a method to find these keywords.

By using this keywords set we can successfully eliminate non-comparative sentences, but the output need not contain only comparative sentences, *i.e.*, using this method one can cover almost all comparative sentences. This clearly shows that it has high recall and less precision. For increase in precision we can generate class sequential rules to further filter comparative sentences.

As for the experiments conducted by nitin on his data sets, only 32% sentences contain one or more of these keywords are genuine comparative sentences. But this keywords are able to capture 94% of all comparative sentences, so 94% recall and 32% precision.

In our experiments the sample results on the I-Phone data file are as below.

I-Phone

Total No of Sentences: 2602

No of Comparative Sentences: 888 (34.1)

No of Non-Comparative Sentences: 1461 (56.14)

For this project, I am just using keywords set to identify the comparative sentences; I am not using any class sequential rules or any machine learning algorithms to identify the comparative sentences. I am not missing any comparative sentences by using this method; I can filter the comparative sentences by using the product name, feature name in the later part of the project. The main algorithm of this project just deals with the comparative sentences having the product name or feature name of the product. If those features or products are missing in the given sentences, then we do not consider them as comparative set. So we are not answering Anaphora Resolution in this algorithm i.e., we do not consider the sentences which give reference about the product or feature in the following or before sentences.

3.2.2 Alternate Methods

The above discussed keywords strategy may be used as a first step to eliminate the non-comparative sentences. This strategy has less precision; in order to increase the precision nitin jindal constructed a database using the words which are within the radius of 3 of each keyword in the sentence.

Then we generate class sequential rules on this database, rules such as finding conjugates such as *whereas/IN, although/IN* etc. in a sentence along with the keyword indicates that it has high probability for being comparative sentence. Similar type of rules was used to filter comparative sentences. This strategy increases precision.

Along with this naïve Bayesian classification model can be used on both keywords and also on class sequential rules, as usual naïve Bayesian classification (Bing, 2006) (Tom, 1997) when used on class sequential rules has more precision than when used on keywords.

The below table shows how different approaches when used on keywords effect the F-score, precision and Recall. (Fig 3.2) Using keyword strategy we do not miss any comparative sentences with high recall, so we apt that strategy. The remaining strategies such as naïve Bayesian, CSR helps in increasing the precision and standard recall. According to the study by Liu, these approaches when applied on different types of data, show steady recall but vary in large scale with precision because *articles* may have long sentences, and for *reviews* classifier may not recognize the comparative sentences, the reason may be due to very small sentences which may not be satisfied by any patterns in CSR. Especially with data from *reviews* that we are dealing in this project, the CSR and NB effect even the recall because the reviews mostly contains non-gradable sentences, which are always hard to filter. So we stopped at initial step i.e. at the step of using keywords, when there is no major impact or no major filtration of comparative sentences by using NB or CSR, its better we do not use them and filter the sentences using other methods.

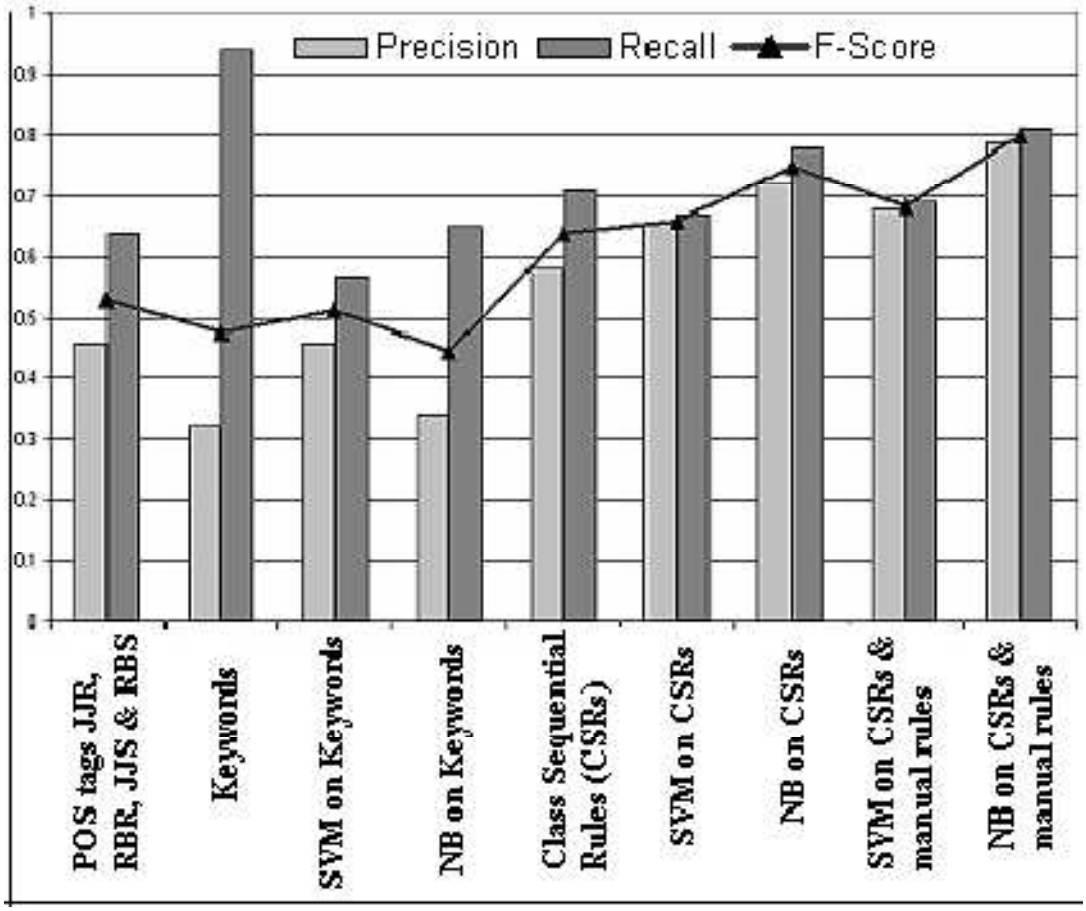


Figure 3.2 Precision, recall and F-score values of different approaches for the problem

(Jindal & liu, August 06-11, 2006)

3.3 Feature Selection

For each product there are many features, and the customers usually express opinion on these features, so to know the opinion of a customer on a particular product we should summarize his opinion on all features of that product. Feature selection has many advantages, they help in (Wikipedia, 2010)

i) Enhancing generalization capabilities

ii) To speed up learning process

iii) Also improves interpretability

Generally people used to find these features in many different ways depending on the domain they work. But for generality POS tags can be used to find these features. Here I am using a Feature set to identify features in the reviews and to summarize the results of entire review. I am not using POS tags to identify the reasons for not using are explained below by explaining the strategy.

3.3.1 Domain Independent Feature Selection

Parts of Speech tagging are used to identify the features because all features are mostly noun or noun phrases. Minqing Hu and Bing Liu in "Mining and Summarizing Customer reviews" (Hu & Liu, 2004) have used POS tags to identify features and phrases. As mentioned earlier we are treating noun and noun phrases as features, and then we identify the frequent feature. We use frequent features because all noun tags need not be features but features are those which are discussed many number of times at least 1% of total reviews sentences. This 1% is minimum support. For general summarization technique i.e., for any domain we can use noun tag /NN as feature and /NN /NN as phrases.

Ex:

> Multi-tasking music player is also a plus when you're on those bored days with nothing to do besides texting.
>
> Multi/NNP -/: tasking/NN music/NN player/NN is/VBZ also/RB a/DT plus/CC when/WRB your/PRP$ on/IN those/DT bored/VBN days/NNS with/IN nothing/NN to/TO do/VB besides/IN texting/NN

The probable features and phrases in the above sentence are

Features: tasking/NN, music/NN player/NN, nothing/NN, texting/NN

Phrases: music/NN player/NN

These are probable features and phrases, the real features are identified by identifying frequent features using association mining. (Liu, Hsu, & Ma, 1998) Then run the association miner based on apriori algorithm on the noun/noun phrase set. (Agrawal & Srikant, 1994) I use this technique for all general projects, i.e., if I do not know the domain of the sentence or review. I used this strategy initially for testing the normal summarization of reviews. I found a lot of unnecessary items selected as features, in order to remove the unnecessary items I created a list of features for each domain that I experimented, so that I no need to search for all noun tags or

noun phrases and I even no need to use association mining and pruning. But as mentioned earlier I used this strategy only for initial experiments, just to study the summarization of any type of product reviews.

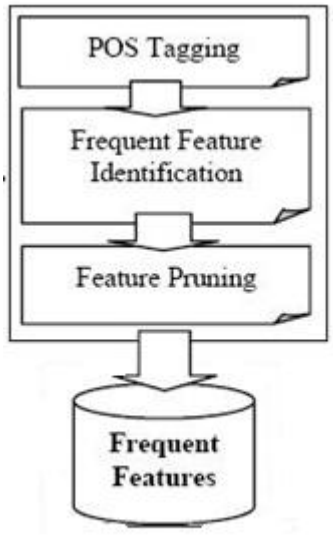


Figure 3.3 Block diagram showing the strategy to identify the features from reviews.

3.3.2 Domain Dependent Feature Selection

As mentioned before I used the above POS strategy for any product reviews, but for this project I limited the reviews only to specific domain i.e., smart phones. So I created a feature set for smart phone domain. Steps to create feature set for any domain.

i) Gather all the features from regular websites, as did for previous method check for features which are most discussed or at least discussed for about 1% of sentences.

ii) Find synonyms for each feature using Word Net.

iii) Find other words may be short forms that are used to mention a particular feature.
Example:

messaging, text

Apps, Applications

iv) Find other sets that effects the polarity of the product.

Example: consider below set of mobile carriers.

Carrier | AT&T | ATT | Sprint | Verizon | tmobile | t-mobile | ALLTEL | US.Cellular | US cellular | UScellular

In the above way we create feature set and check for that particular feature and polarity of the product with respect to that feature in order to summarize the final result.

The main advantage of this method is

i) We never ever again see unnecessary items polarity and therefore reduce the burden on the system.

ii) No different approach for features and phrases anything can be searched if inserted in feature set.

iii) Increases the efficiency when dealing with large database because we no longer need to find the polarity of non feature items. In the previous approach by Bing (Hu & Liu, 2004) we used to identify polarity of each feature and then identify the feature frequency and after pruning we decide whether the item is feature or not, but here we know what is feature and what not is feature so there is no need of association mining and pruning.

iv) Need to do POS tagging, no change with previous approach because to find the polarity we need to identify the adjective.

There are many disadvantages along with the above advantages, they are

i) Need to create separate feature set for each domain

ii) Need to update the feature set when there a new feature added to a product.

Even though there are disadvantages, I prefer this approach because the task to prepare the feature set is a onetime process and to add new feature to feature set is not so difficult as the program for this project is automated in such a way that the new feature or introduction of new product does not need to do anything with coding.

3.3.3 Feature Sets

The summarization of product is based on the summarization of features, but will all features have equal effect in deciding the polarity of a product, is all features are necessary in deciding the polarity of a product. The answers to the above question are answered in this section.

While deciding the polarity of each product, many factors need to be considered, the first and most important one is feature ranking. For this I created three sets A, B and C.

i) The set A contains the most important and prominent features which can change the polarity of a product by a great extent.

ii) The set B contains less important features but these features can affect the polarity of a product to some extent.

iii) The set C contains unimportant features but are features as they are part of product, the positive or negative polarity of these features will never have much effect on the polarity of the product.

I followed the steps below in order to identify the most and least important features of a product.

i) Collected all the domain expert opinion on the features of the product.
Examples: CNet, Consumersearch, and Amazon for any electronics.

ii) After collecting the domain expert opinion a initial ordering will be formed, as discussed below.
The domain expert opinion does not cover all features, because they just give review on only few features, which are common to all products. The below presented domain expert opinion is for smart phone domain.
Each feature has many sub features which are shown in figure below (Fig 3.3 (b))

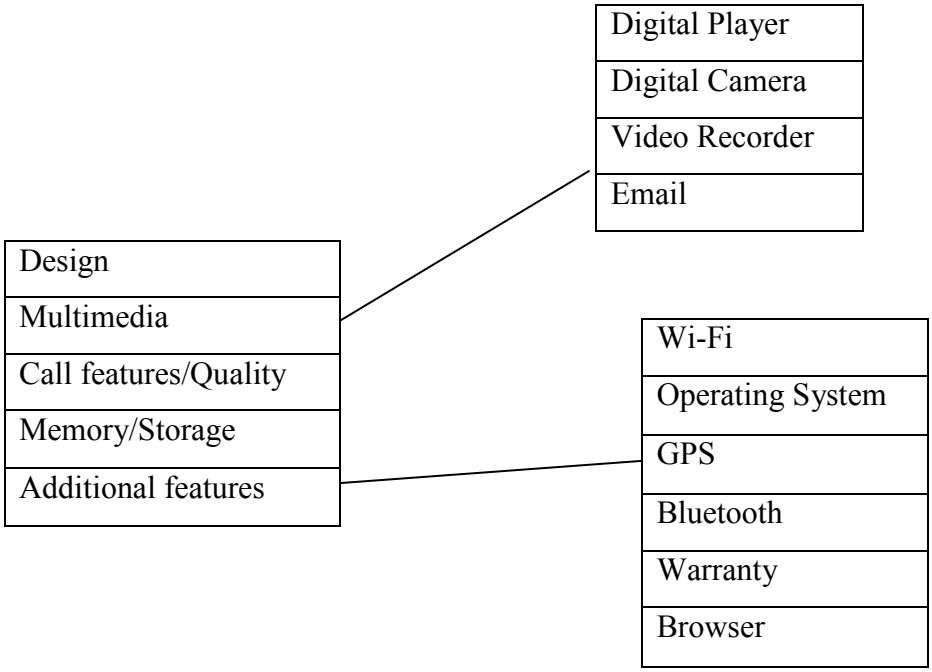


Figure 3.4 Domain expert ranking of features for smart phone domain

iii) After taking domain expert opinion then use the data to decide the ranking of the features. Check for the count of each feature, it shows how many times the corresponding feature has been discussed. We assume that the feature discussed more is the featue required more.

iv) Not only in our data set but also by using google fight (Nation & Waring, 1997) Where the two corresponding feature count values can be obtained from the large web-based corpus.

Using the above steps all the features are distributed into the three sets A, B, and C. We store features along with their synonyms and alternate names. The figure below (Fig 3.3(c)) shows the steps involved in deciding the set for each feature.

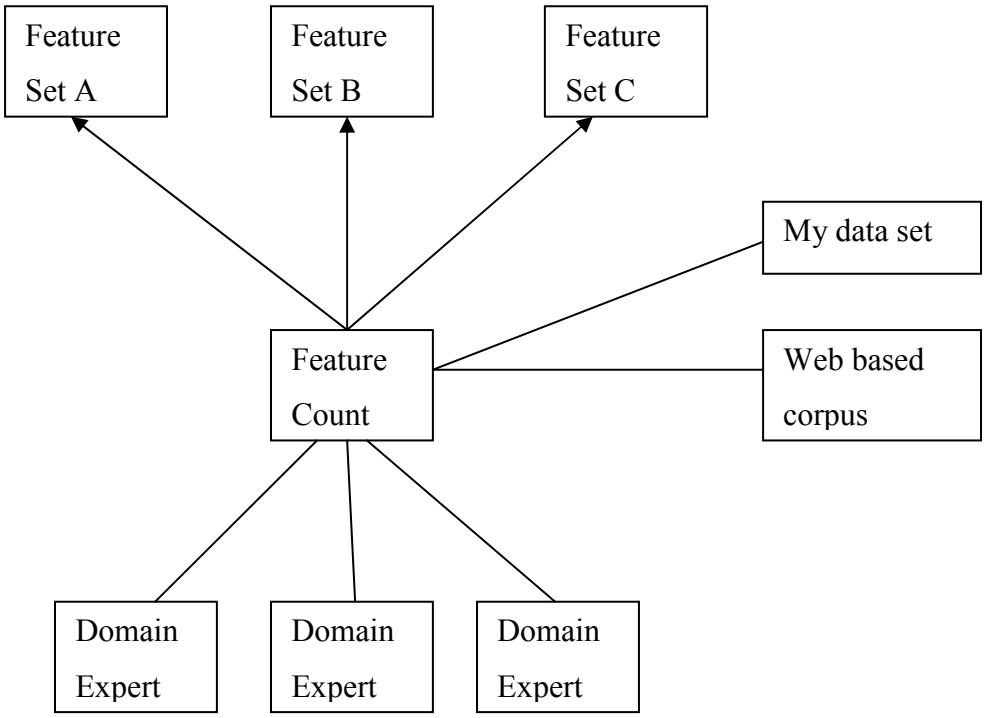


Figure 3.5 steps involved in deciding the perfect set for each feature

3.3.4 Scores to Feature Sets

Now we have three sets A, B and C with features. Now we need to give corresponding scoring to each feature set. To give score or weight to each feature set we depend on the **tf–idf** weight which is term frequency–inverse document frequency. This is a statistical measure to evaluate how important a word is to a document in the collection. (wikipedia, 2010).

Here each set is given separate weight instead of each feature. We consider that all features in a set have equal importance, so a common weight is given to those features. The weight of each set is depend on the number of times the features in the set are repeated.

In this term frequency–inverse document frequency, the term-frequency is calculated as below.

Consider 100 words in a document and a feature display appears 25 times in this document then

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$$tf = 25/100 = 0.4$$

Now suppose if we have 10 documents and if the same display feature appeared 200 times in these documents then inverse document frequency is calculated as

$$idf = \log(10/200)$$

tf-idf is the cross product of tf and idf i.e.,

$$0.4 * \log(10/200)$$

In this way we can find weight of each feature, but here we need to find the weight of entire set. The strategy that we are using is we counted all the feature counts in set A and also in set B. I found the percentage change in count between A, B and C and then assigned the weightage based on that difference in percentage change between all three. I myself restricted to give 1 as maximum weightage and 0 as minimum weightage so each set will have weightage between 0 and 1.

3.4 Determining Polarity

In order to determine the polarity of anything, we need to identify the opinion words. Opinion words are those words that are used to express subjective opinion. Rebecca has proved that presence of adjectives is useful for determining whether a sentence is subjective. (Bruce & Wiebe, 1999) (Wiebe, Bruce, & O'Hara, 1999). So we use adjectives as opinion words and we also extract opinion words from sentences having at least one feature.

The modified algorithm from bing is shown below

Algorithm 3.4 : Opinion word extraction

```
for each sentence in the review database
        if (it contains a frequent feature, extract all the adjective
           words as opinion words)
           for each feature in the sentence
           the nearby adjective is recorded as its effective opinion.
```

Now once the opinion words are identified we need to identify the semantic orientation of the opinion word i.e., adjective. Wiebe proved that there are adjectives without any orientation. (Hatzivassiloglou & Wiebe, Effects of Adjective Orientation and Gradability on Sentence Subjectivity., 2000). There are many tools available to identify the orientation of a sentence but I used Jane16 open source tool.

3.4.1 Jane16 Methodology

Jane16 is an open source tool providing best sentiment analysis algorithm. The reason for choosing jane16 is due to its huge training data and that too it has been trained heavily on product review data set, which we are dealing in our project. The data set wschema consists of two files wschemaNegative and wschemaPositive, these csv files consists of negative and positive words along with their unique id and weight of the phrase. This database is build by reading online reviews sites positive (one with 5 stars) negative (with 1 star).

Jane16 is a pure statistical engine, (marianmedla, 2007) with no lexical analysis i.e., we are not using any nouns or verbs or any POS tags while determining the polarity. The opinionated word is sent to database and searched in both positive database and negative database and the result is one with highest score.

Example: I hate it

This phrase is present in both positive and negative database. The score for this phrase in corresponding csv files are as below.

In negative csv: I hate it, 1-16-1921716, 318.45
i.e,. it has 318 hits in database with negative polarity
the middle one is the unique id

in positive csv: I hate it,1-48-714060, 66.45
here it has just 66 hits in the database.
So the phrase is negative.

The database is developed by crawling the web, statistically computing references and occurrences. Here the concentration is more on the word occurrences. While preparing database synonyms to the items in the database are found using Word Net.

Algorithm 3.4: Determining semantic orientation of opinion word

```
1. Procedure OrientationSearch(adjective_list, wschemeNegative, wschemePositive)
2. begin
3.        for each adjective wi in adjective_list
4.        begin
5.            if (wi has synonym s in wschemeNegative)
6.                { wn's weight = s's weight;}
7.
8.             if (wi has synonym s1 in wschemePositive)
9.                 { wp's weight = s1 weight; }
10.            if (wn's weight > wp's weight)
11.                wi's orientation = negative orientation
12.           else
13.                wi's orientation = positive orientation
14.      endfor;
15. end
```

3.5 Template Generation

The following two output templates are generated for the methodology defined. These two templates are generated for one product reviews. One template shows the corresponding feature evaluation and second template shows the evaluation of alternate product when compared to this product with respect to feature.

1) <Feature, Feature Count, Positive, Negative>
   - Feature : Features of a product (example: Display, Wifi, Bluetooth, Screen Size)
   - Feature Count: Number of time the feature is discussed in the product reviews.
   - Positive: Subset of Feature Count, gives total number of positive feature instances in the product reviews.
   - Negative: Gives total number of negative feature instances in product reviews.
2) < Product, Feature, Feature Count>

   This template is generated in parallel to the above template. This template is generated for each product.
   - Product: This is an alternate product in the review for the sentence where a particular feature of a product is compared.

Example: suppose we are analyzing the reviews of *I-Phone* and we encountered a below sentence in one of the review.

“ I-Phone has a better display than EVO”

For this sentence the template is

<EVO, display, 1>

The second template is mainly for comparative sentences, which shows all the compared products, along with the features that they are compared.

3.6 Analysis of Results

The results are analyzed from the templates generated, The initial results just show the evaluation of a single product. The final results are generated by summarizing the results from all the products.

The expected final results are.

1) The best product for each feature
2) Top features of a product.
3) Best Product in a domain.
4) Summarization of a single product by considering all the reviews of another products.
5) Best alternative product for a product that you are looking for with respect to feature.

To find the above results the below steps are followed.

1) To find the best product of each feature.

   Gather all templates of all products and separate them with respect to feature. Then the best product is the one which has more profeaturecount value. This value is calculated using the product X own reviews and its positive and negative count for a particular feature in other product reviews.

Profeaturecount = Positive count of features $- \sum_{i=1\ to\ 10} NC_i + \sum_{i=1\ to\ 10} PC_i$

Where NC = negative count of this product for feature in other product reviews.
PC = Positive count of this product for feature in other product reviews.
i = products (example: I-Phone, Evo, Droid X etc….)

An example by quarkrank shows the similar feature in which they did not consider the comparative sentences i.e. they did not consider the negative and positive effect created by reviews of other product.

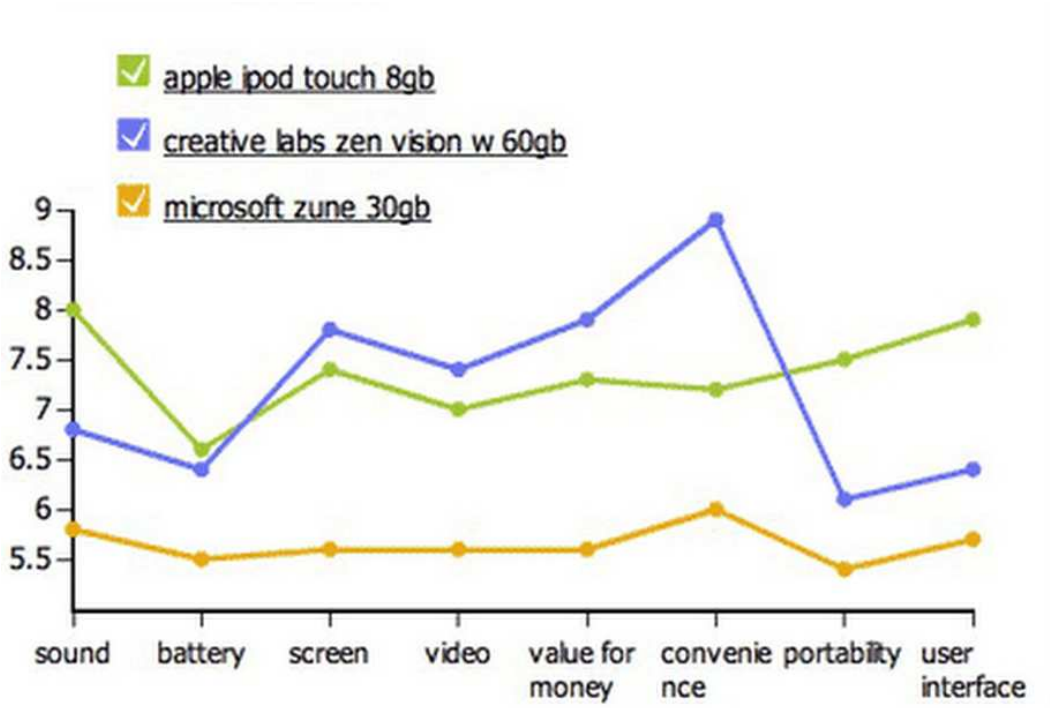


Figure 3.6 Figure showing the best product of each feature.

2) Top features of the product:

To find the best features of a product X, we need to analyze the template 1 of product X and also template 2 of product X in other product reviews.

A particular feature can be very good in this product but when compared to other product it may not be the best.

Top features of a product X are those which have highest positive count after subtracting the negative effect created by other product on the features of product X.

Figure 3.7 Figure showing the top features of a product ipod.

3) Best Product in the domain:

This analysis requires feature scoring which I discussed previously. For determining the best product I consider the positive count of all features of a product and give a scoring to the product based on the feature. There may be some features which are not so important but are discussed more and may be getting more positive count because of that feature. So we use feature scoring to eliminate this.

Example:

In the above Fig 3.6(a), the user interface and portability has highest weightage, than screen, video, convenience so apple ipod is better product than creative labs zen though the later has more positive counts for features.

4 & 5 can be obtained from 1 & 2. Alternative product for a feature can be obtained by choosing the next product in the top products of that feature. Similarly for summarization of a product, we find top features of a product by summarizing all its features.

3.8 Summary and Discussion

In this section I presented the novel summarization approach for user reviews using comparative sentences and feature weight-age. We first separated comparative sentences from normal sentences using keywords strategy. Then I extracted features, for this I approached two ways one is domain independent feature selection for initial experiments and the other is domain dependent feature selection for main experiments. Then I create feature sets, for giving proper weight-age for each feature. Finally I determined the polarity of a sentence using the database created by jane16 group.

Chapter 4 - Experimental Setup

In the previous section I explained the methodology in general sense, now in this section I give the experimental setup, used for this thesis. We have a smart phone database with 11 smart phone reviews to improve the methodology of Bing Liu and to diversify the summarization results. The diversification of results is done by feature and product based diversification.

The domain chosen for this project is based on the 3 steps.

1. Domain which can be expandable to large database.
2. Domain where results can be comparable, i.e. where we can match our results with domain expert's views.

4.1 Data Collection

For this project I collected data for about 11 leading smart phones, (Customers, 2010) (Customers, Cell phones, 2010) using Various Sources such as crawling and manual collecting. I collected data for these smart phones from Amazon and Cnet. The problems while collecting the data set are.

1. What should be the size of each smart phone data, in other words how many reviews should be collected for each smart phone?
2. From where we should collect reviews?
3. What data should be collected along with product review, i.e. reviewer name, the rating that he gave etc?

```
4.5 stars
"Great product but with some misgivings" on
July 1, 2010 by rdekoch (10 reviews)
Pros: Like butter! Fast performance, awe-inspiring display,
camera is pretty damn sweet,battery life is much
improved, super sexy design

Cons: Antenna issue that I don't need to explain. arrogant
public responses to antenna issues, glass scratches
more easily than one would expect, AT&T's coverage
in San Francisco SUCKS

Summary: I've tried to focus on hardware in my review but
the hardware in concert with ios 4 is a very substantial
and welcome upgrade. I really like using this phone for
gaming, texting, email and web surfing. I was initially
sent a deffective phone and had to jump through some hoops
to have it exchanged (although they were very professional
and nice). That coupled with the antenna issue made me resist
liking this phione. It seemed like a lot of hassle for little
improvement. After using the phone for a few days, I have
come to really love it. The camera and display alone make
it a worthy upgrade. After a rocky start, I am very happy.
```

Figure 4.1 Example of a review document from dataset

The size of the product data is always an important problem because the size affects the polarity of the product.

For example: Consider that we taken all equal number of reviews for all 11 smart phones say it as 100. We will not have any problem if we treat each phone separately, but in this project we are considering comparative sentences, so when we consider comparative sentences, and when there is something negative we should give negative scoring to that product.

Most of the smart phone reviewers will try to compare their favorite product with market leader or leader in their domain. Leader in their domain means, leader in mobiles using the same operating systems. With this the leader when ever compared gets more negative rank

than the number of positives in its review. This should be compensated otherwise; the phone which has less negative reviews or fewer comparisons gets benefited.

The present market leaders order is given below.

1	I-Phone 4	
2	Droid X	73
3	HTC EVO	204
4	Blackberry torch 9800	102
5	Droid incredible	224
6	Samsung vibrant	124
7	Motorolo droid 2	8
8	Samsung epic 4g	82
9	Htc droid eris	36
10	Palm pre	100
11	Samsung omnia	34

Table 4.1 Smart Phone market Leaders, and number of comparisons with market leader

(Smartphones Review, 2010), (best smartphones, 2010)

Solutions to the above problem can be

1. Compensating by taking more reviews for market leaders or in other words for which there are more comparisons. But how much ?
2. Other strategy that I used in this thesis, is taking the positive percentage of total occurrences of that feature.

The strategy that I used is taking the percentage of positive occurrences of each feature.

For example: If I-Phone 4 is has 45 positive and 20 negative instances about its display in I-Phone reviews and 15 positive and 25 negative instances in other reviews. I take positive percentage of I-Phone 4 display instances i.e.

Total positive = 45 + 15 = 60

Total negative = 20 + 25 =45

Percentage of positiveness = 60 / 60 +45 = 0.571

In general it is return as

$$\text{Score} = \frac{\sum TP}{\sum TP + \sum TN}$$

The data should be collected from a trusted review site such as amazon or cnet because, there can be a scope of fake reviews. These trusted websites block the fake reviewers and repeated reviewers.

Along with review the other information to be collected is

1. Star rating by the customer (if any)
2. User id of the reviewer in order to store in database and eliminate duplicate reviews from the same user.
3. Date of the review, which plays a major factor because, by the date he write reviews all the products that we are comparing should be released.

In this way we collect reviews for all the 11 smart phones and each review is stored separately in a text document.

4.2 Automated Polarity of Reviews

After storing each review in separate document we find the polarity of each feature in the review of a smart phone using the methodology defined in section 3.4.

Before finding the polarity of features, we divided all features of a smart phone into 16 sections. I accommodated all the features of smart phones into these 16 sections.

For example consider the section below

Display: "display", "brigtness", "brighter", "bright", "brightest", "screen", "touch"

If any of the above 7 feature is found it is considered as display feature, i.e. we find the synonyms and alternate words of the all the 16 features and make them as section.

Alternate words such as “reception” and “call quality” both represent the signal strength, *i.e.,* if signal strength is weak call quality will be weaker. Now each review of a product is passed

into the system for identification of features, and once features are identified we find the positive and negative polarity of each feature by using the sentence of that feature.

First we find the polarity of each feature using wschema which has positive and negative database, and after finding the positive score and negative score of each feature, we accumulate all features into the 16 sections.

The template generated for a review is shown below:

4,3,?,4,4,3,4,4,?,?,4,4,?,?,?,?

The 16 sections for this smart phone are.

- Display: The display section may include features like screen, touch, brightness
- Camera: { megapixel, camera, digital zoom }
- Storage: {Memory, storage….}
- Battery: {battery, battery life….}
- Multimedia: {music, pictures, ringtones }
- Web: { Internet, 4g, 3g, browser, }
- Email: {mail, message, gmail, email }
- Keyboard: {type, swype, keyboard,}
- Reception: {signal, dropped call, call quality, reception }
- OperatingSystem: {OS, IOS, Android, etc..}
- Applications: {Apps, market, appstore, widget, appworld }
- Bluetooth: {Bluetooth}
- Processor
- Flash
- Navigation: {gps }
- Carrier: {ATT, T-mobile, etc…}

The above template just shows the rounded positive percentage score of each feature.

I created this as instance and the class label is given manually for each review. In this project I am verifying with manual annotation for just three smart phones I-Phone, Droid X and EVO.

For each of these phones I created instances automatically for each review and attached class label for them.

The above instance shown has values that are adjusted between [0 – 4]. The values are adjusted in such a way that if there is any negative percentage or if a particular feature has more negative count in a review it is given as 1.

0	Worst, Poor, Terrible, far inferior to
1	Fair, Not as good as, inferior to
2	Ok, Good, All right, not bad
3	Very Good, Better
4	Best, Excellent, Much Better

Table 4.2 Rules to rank each review

The automated experiments initially give results with template of more than 72 features because of various synonyms used for each feature. I scaled down them to 16 features by summing all the synonym features results into one.

I made sure that the class label does not have decimal numbers such as 3.5, 2.5. By having class labels in decimals I got very bad and horrible results. The reason for not having decimal class label is if I have decimal class label, the weka (I & Frank, 1999) system is in dilemma to give value to instance when it found result close to 3 and 3.5 and it preferred 3 most of the time than 3.5 and I always ended up with 0 predictions for 3.5.

4.3 Evaluation - Experiments

For evaluating the automated results generated to decide the polarity of whole document with respect to feature by considering the comparative sentence, I annotated the all the reviews of products and generated template for each review in a product.

Here I used just 3 smart phone reviews for evaluating. So to find the review score or average score of each product one and to find the top features of a product we evaluate automated data with manual data. The top features of a product can be found using feature selection *i.e.*, to select subset features which effect the ranking of the review or whole product.

There are many algorithms that are used to select a subset of features in one of the three ways:

1. Filters
2. Wrappers
3. Embedded

Here we use wrappers to determine the subset of features. The papers by kohavi and Hsu et al (R & G, 1997), (Hsu, Welge, Redman, & Clutter, 2002) explained genetic algorithm approaches to perform attribute subset selection. They suggested that wrappers increase prediction accuracy. The wrappers have a problem of overfitting with small training set (Sanmay, 2001). Here we use wrappers instead of features because according to Kevin Dunne wrappers will outperform filter-based approach to feature selection where an adequate amount of data is available (Dunne, Cunningham, & Azuaje, 2002).

The procedure to generate feature subset is 3 types

1. Complete: search such as best first, beam search, exhaustive search etc… falls into this category.
2. Heuristic: sequential forward selection, sequential backward selection etc… falls into this category. Sequential forward selection is something where we take small subset increase it and decide the final subset, sequential backward selection is done in the other way by initially taking all the attributes and later removing one by one till a proper subset is found.
3. Random: Genetic algorithm, random generation, simulated annealing etc are in this section.

   (Prasov, 2008)

In this thesis we evaluate the subset of features generated by wrapper with the top features of the product from the automated system.

4.3.1 Evaluation with Domain Experts

Before evaluating the automated data generated by system with manually annotated data, I will also evaluate both the manual and automated data with the data from the domain expert. Why do we need to do this?

The reason for doing this both the automated and manual data may not have proper weightage and research on each feature which is done by domain experts. The domain experts does an exhaustive research on each feature and give proper weightage for each feature, which is similar to our task which we did here using the reviews of product by users. If we could match the results with domain expert result we can save a lot of research funds which is used by domain experts on this research.

But the availability of data from domain expert is too less and to train the system with such a less data is always difficult and expecting good results with such a less training data is far from possible. Here I gathered domain expert data from a website where they have done a exhaustive research on this smart phone field (Smartphones Review, 2010) and provided scoring to each smart phone and also for their features form range [0-4].

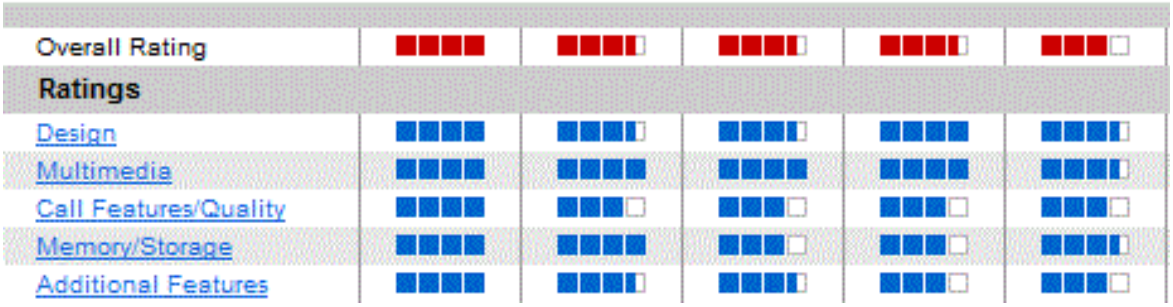


Figure 4.2 Score for each feature in smart phone by domain experts

The above figure, (Figure 4.3) shows you the rating given to each feature by the domain expert. Here according to domain expert he gathered all feature into these 5 sets. Each has subsets just like design has subset features such as phone style, display resolution, screen, *etc*.

With all these feature scoring we had created arff (Waikato, 2008) file to train to weka. We have data for just 10 smart phones, which is very less to train and then we validate our automated and manually generated instance file (ARFF) with this data.

Chapter 5 - Results

Having discussed a methodology for comparative sentiment analysis in Chapter 3 and experimental setup in Chapter 4, in this Chapter I will present the results of some initial experiments and of applying the given methodology.

5.1 Results from Preliminary Experiments

The preliminary experiments are conducted on all 11 smart phones using my methodology of using comparative sentences and feature selection from a quite popular algorithm by Bing Liu (Hu & Liu, 2004).

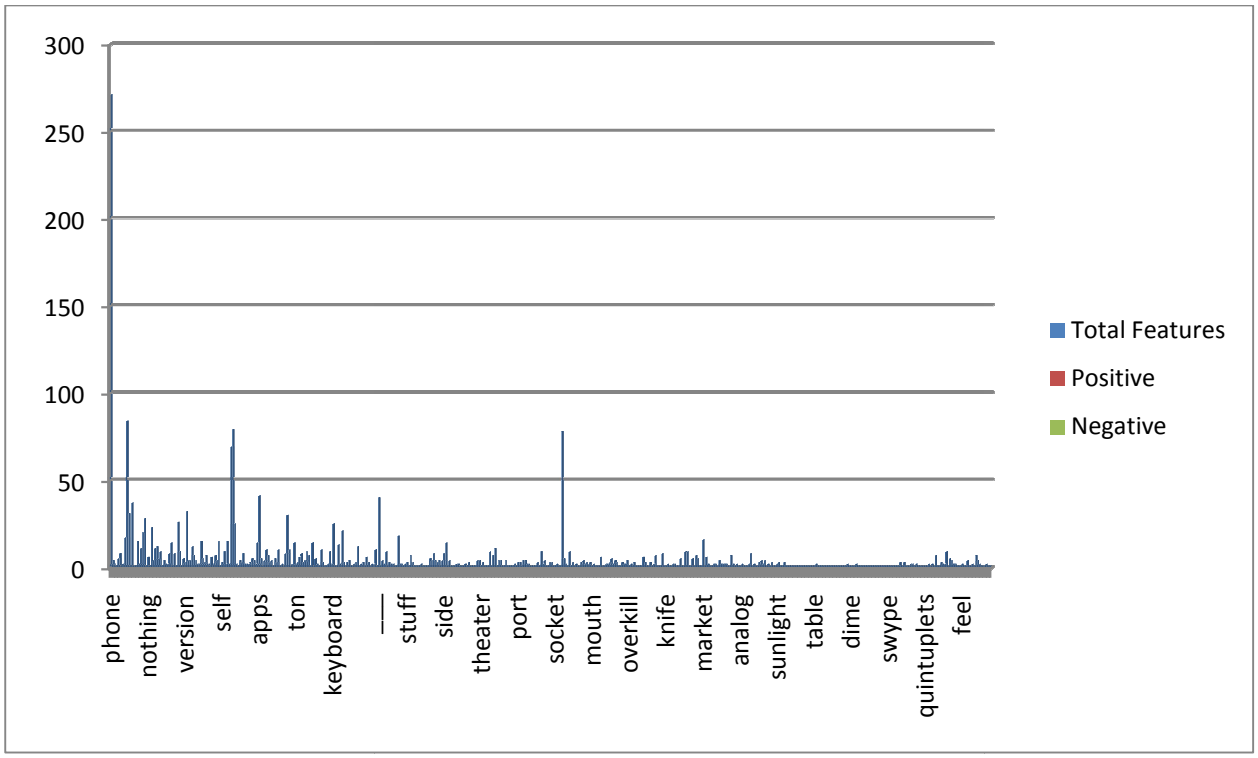


Figure 5.1 Results showing the number of features before using apriori algorithm.

(Wiki, 2008), (books, 2010)

The above Figure (Figure 5.1 (a)) shows you the number of times each feature has been appeared for i-phone data set. As discussed earlier in Chapter 2 and Chapter 3 here we follow Bing Liu, methodology to determine feature, *i.e.*, we are using /NN tag for feature. But if you

observe the graph many of the unwanted words such as “self”, “ton”, “stuff”, “side” are used as features. On a whole using this method we got around 501 features for I-phone where the number is far from reality. So now we apply apriori algorithm to minimize or extract feature set.

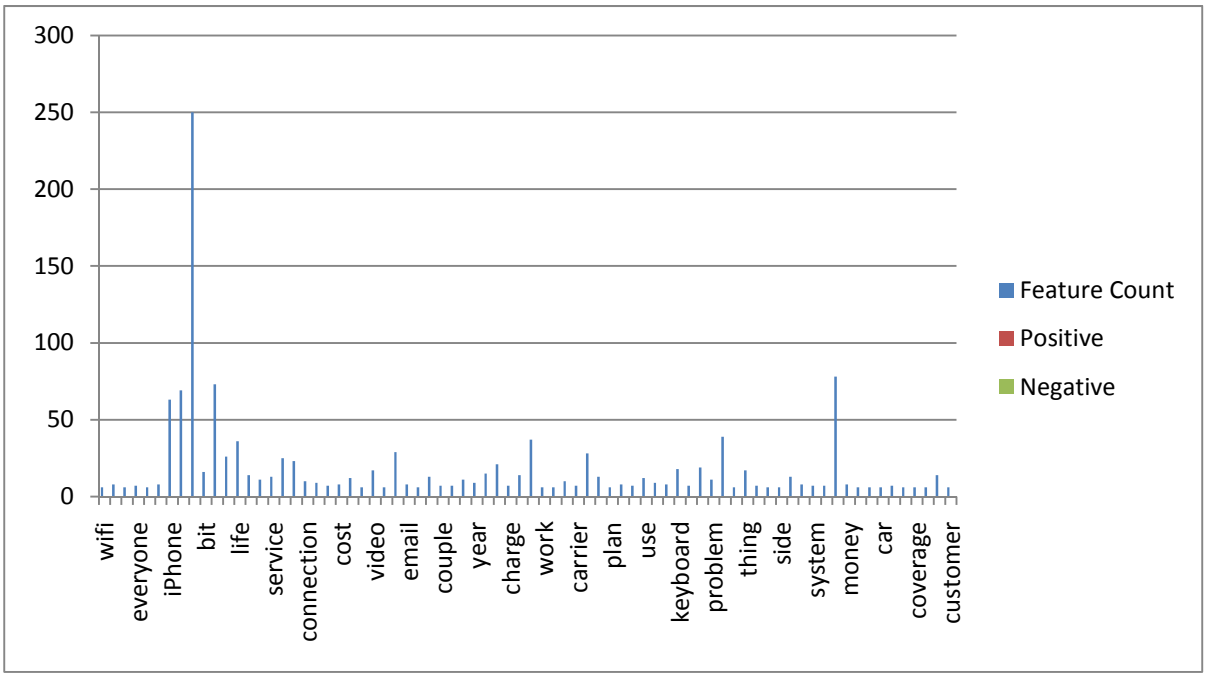


Figure 5.2 Features and feature count after applying Apriori algorithm

If we observe the above diagram the features obtained make sense, here we applied apriori algorithm, to remove the unnecessary item which are called as features in the previous step. We kept the count as 6, here I assumed that a feature will be discussed at least or more than 6 times and removed those which not even had a count of 6. Then we found the opinion of each feature.

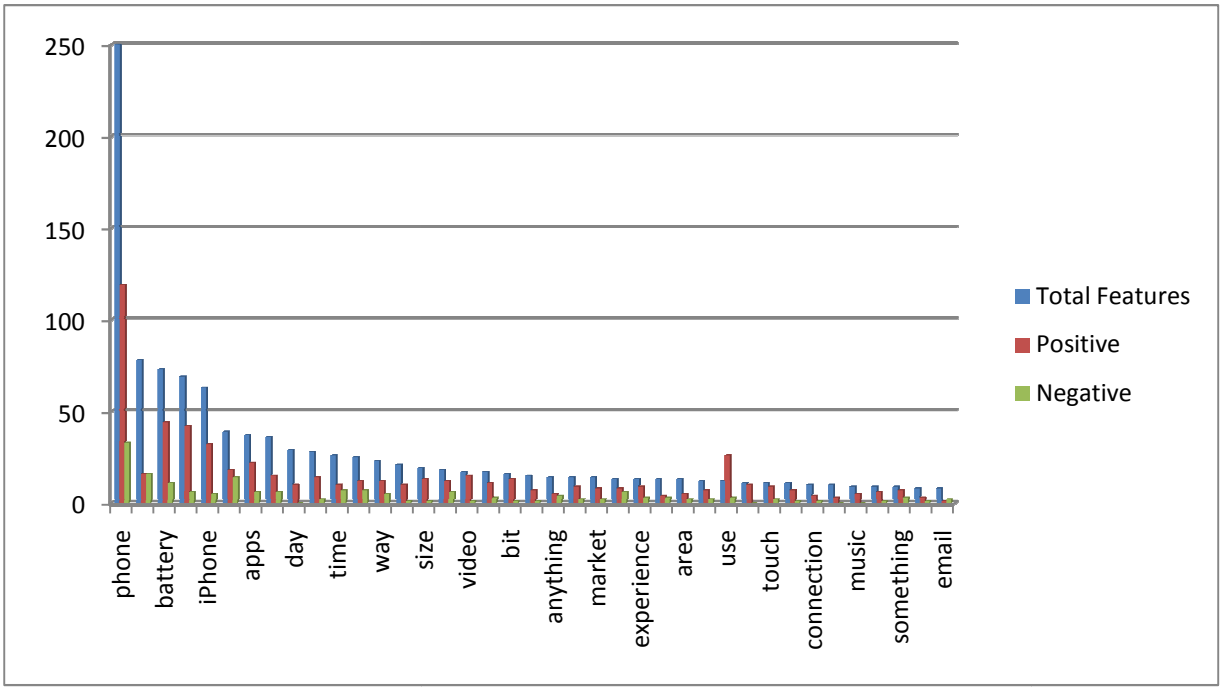


Figure 5.3 Positive and negative opinion count of each feature for EVO

This figure shows the positive and negative count of each feature discovered in the reviews. These initial experiments are domain independent because here we decide what is feature and what is phrase based on POS tags, here we no need to have any domain knowledge. This methodology is proposed by bing liu to use POS tags, and I extended it to using comparative sentences, the score that u see for each feature is not just from the reviews of that product, but also from other product, whenever compared with this product. As mentioned earlier the above domain independent methodology does not predict feature values with 100% accuracy; it predicts some to less than 50% features. which is poor bad. So the next step in this process is domain dependent feature selection.

Scores for the other two phones are given below.

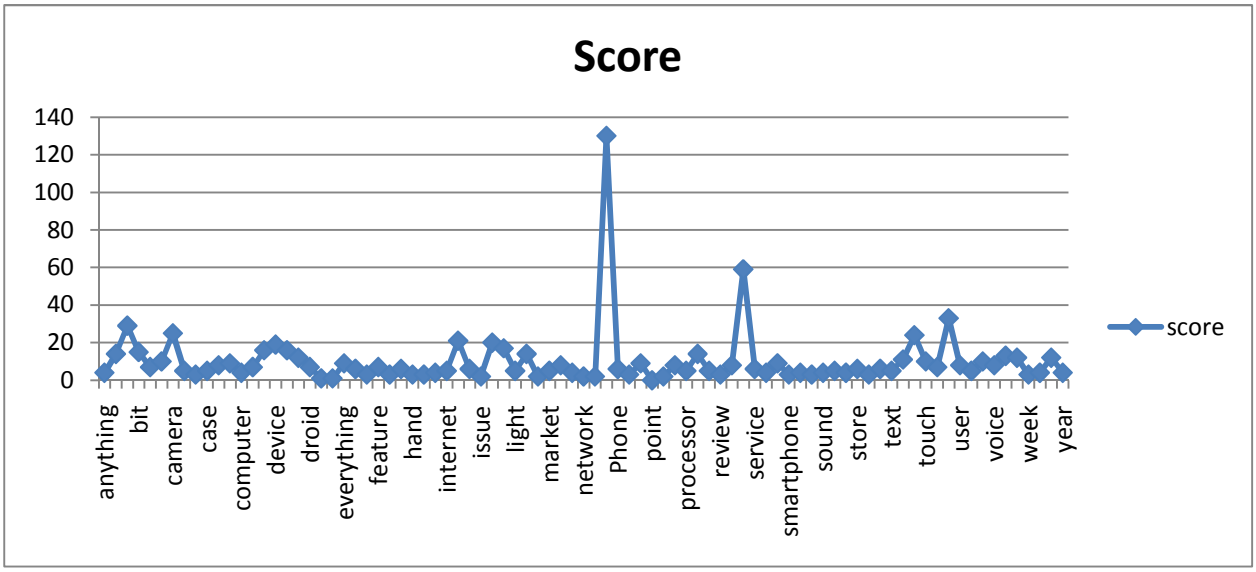


Figure 5.4 Cumulative score for features of Droid X

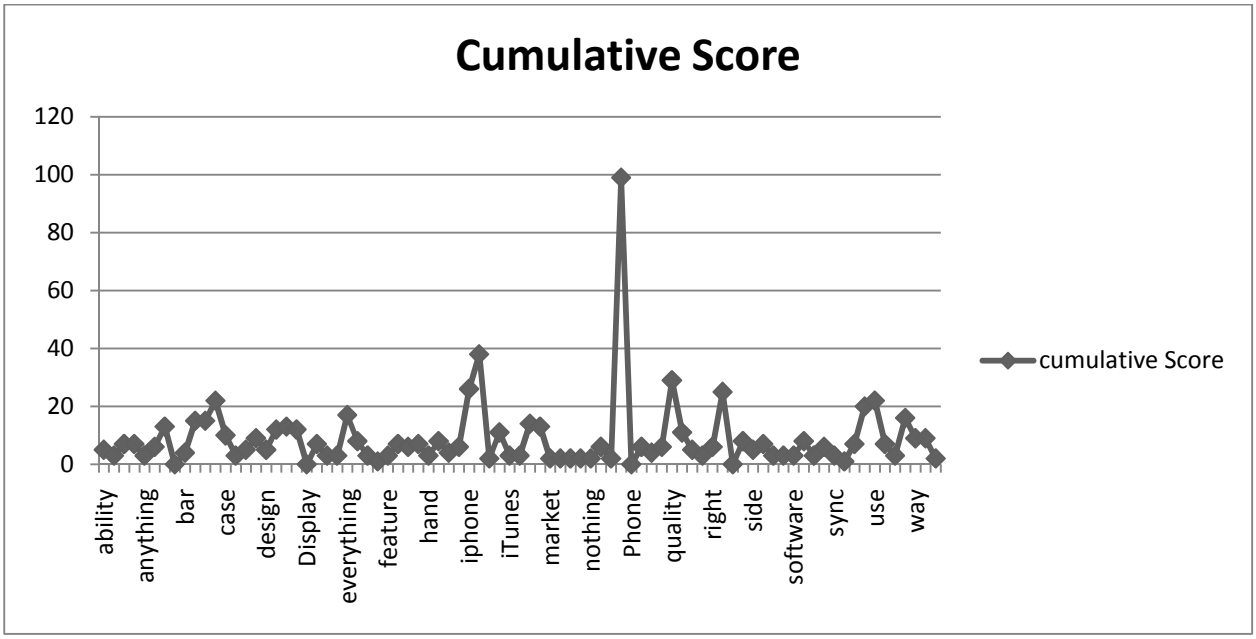


Figure 5.5 Cumulative score for features of I-Phone

If we observe the trend we can easily point out which are the best and worst features of a phone and with more manual work on each data of a smart phone the results can be used to compare with other smart phones. With the results that I showed it cannot be compared because these smart phones does not have 100% common features and the features are given equal

weight-age in determining the winner. So this approach can be used to study about a smart phone individually, i.e., just to have a over look about a product and just to know the trend this domain independent approach can be used. But to completely survey and research and compare product we need to have more detailed work.

5.2 Results for Domain Specific Methodology

The below results are for the same data sets and the using this methodology the smart phone features can be compared to 100% because here we gathered all features to 16 sections and every feature falls into one of these sections and as in previous models we did not miss any single model.

As discussed before the results with 16 sections are shown below.

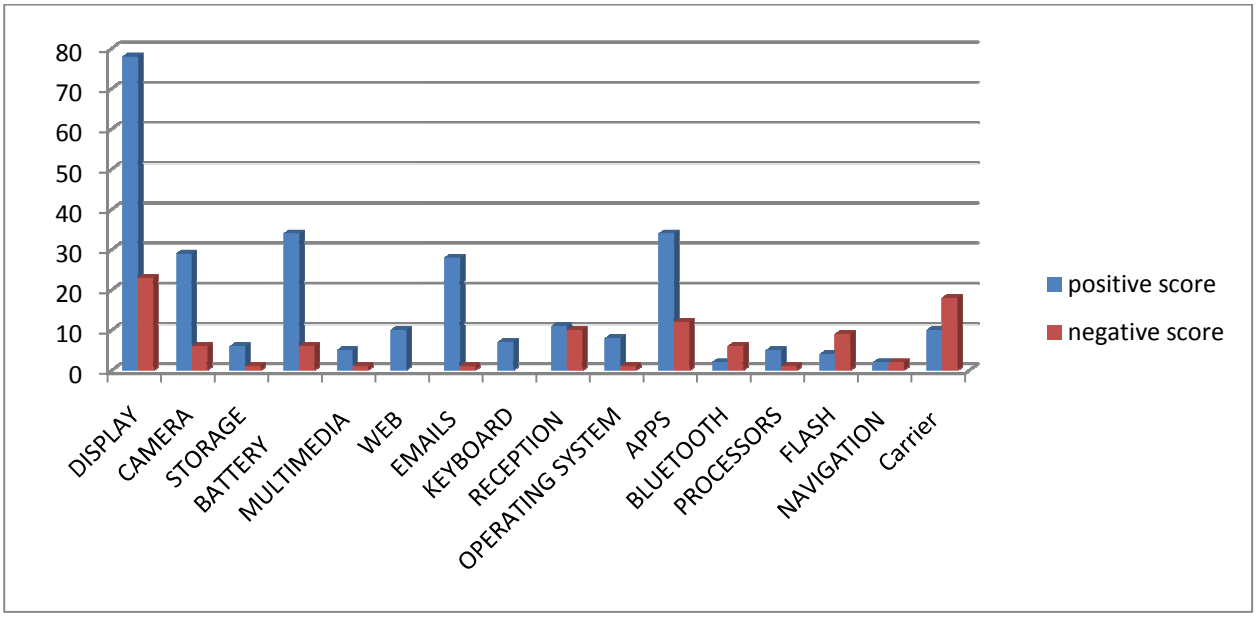


Figure 5.6 Feature sections positive and negative scoring for I-Phone 4

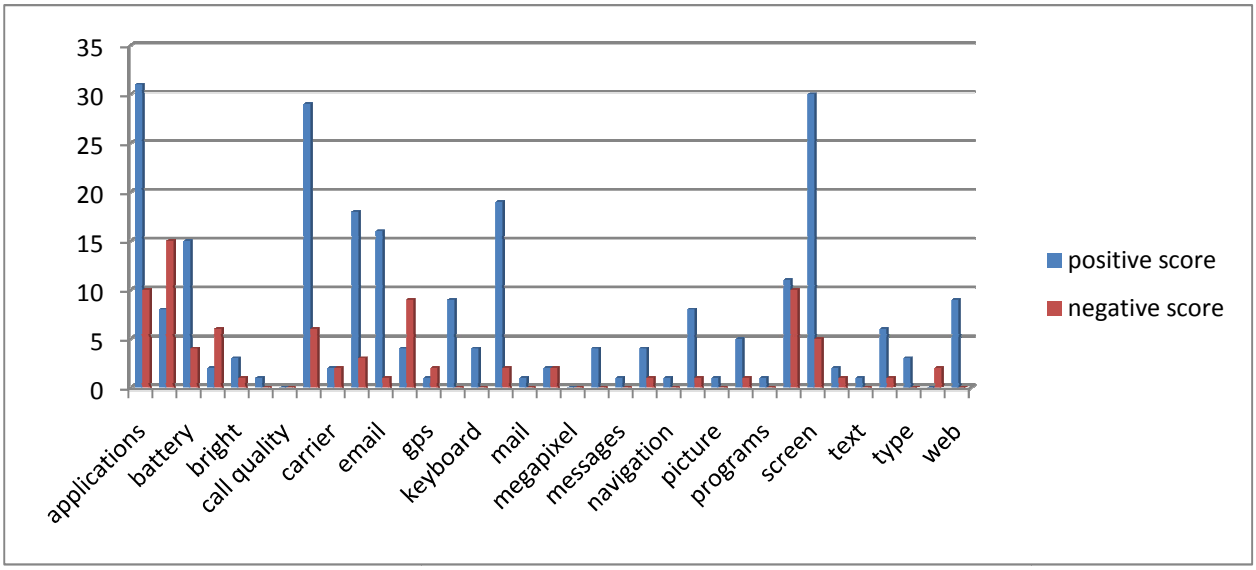


Figure 5.7 Feature positive and negative scoring for all features of I-Phone 4.

The above figure (figure 5.2(b)) shows all the features. The reason for making these features as 16 sections is to compare with other smart phones. The best thing about these sections is we do not have any missing features while comparing.

If we observe closely the above results it is clear that people have problems with carrier, Flash and bluetooth while using I-Phone. They are clearly satisfied with its display, performance and applications. This also shows the top features of the product.

The top features of the product are those with less negative score or more overall positive score percentage.

The top features of each product are given below.

I-Phone	Droid X	EVO
Applications	Screen	Battery
Screen	Battery	Applications
Camera	Camera	Screen
Battery	Application	Carrier
Display	Carrier	Keyboard

Table 5.1 Table showing top features which affect the score of each product

The above shown table for top five features of a product may be quite interesting but those are the features that are most widely discussed either in positive way or negative way. If we observe Battery feature of EVO, we get more negative count of EVO, it just shows how many times the feature is discussed or how much effect that a particular feature has. This is used to decide the A, B, C sections for creating feature score, which we had discussed in Chapter 3.3.4. Using term frequency–inverse document frequency and review from domain expert I decided the A, B, C sections as below.

A	Display	screen	Touch	os	battery	storage	camera
B	app	web	carrier	bluetooth	reception	Processor	
C	flash	navigation	multimedia	keyboard	email		

Table 5.2 3 Sections for Feature Scoring

The feature score for each of these 3 sections is calculated by difference is percentage of each section feature count; *i.e.*, we take total all 3 sections feature count as 100% and calculate the individual score of each sections.

After calculating the individual score is given as below:

A: 0.511

B: 0.35

C: 0.13

These scoring can be applied to any smart phone, to calculate the overall scoring or ranking of smart phones. Using this scoring I calculated score to each smart phone to compare smart phones.

The scores for 3 smart phones I-Phone, Evo and Droid X are

I-Phone: 3.73 (rounded to 2 decimals)

Droid X: 3.53

EVO: 3.62

The rating are adjusted to range between 0 – 4 because the domain experts gave range from 0-4 so that we can compare them with the domain experts. The domain experts ranking is quite similar but not exactly equal.

Domain Experts ranking is as follows:

Ranking	Domain Experts	My system
1	I-Phone 4	I-Phone
2	Droid X	EVO
3	HTC EVO	Droid X
4	Blackberry torch 9800	Blackberry Torch
5	Droid incredible	Droid 2
6	Samsung vibrant	Droid Incredible
7	Motorolo droid 2	Palm pre

Table 5.3 Ranking Comparison of Domain Expert and My System

The ranking is given by proper scoring of each feature and by considering positive percentage of each feature. The above table shows that we almost achieved the ranking given by domain experts.

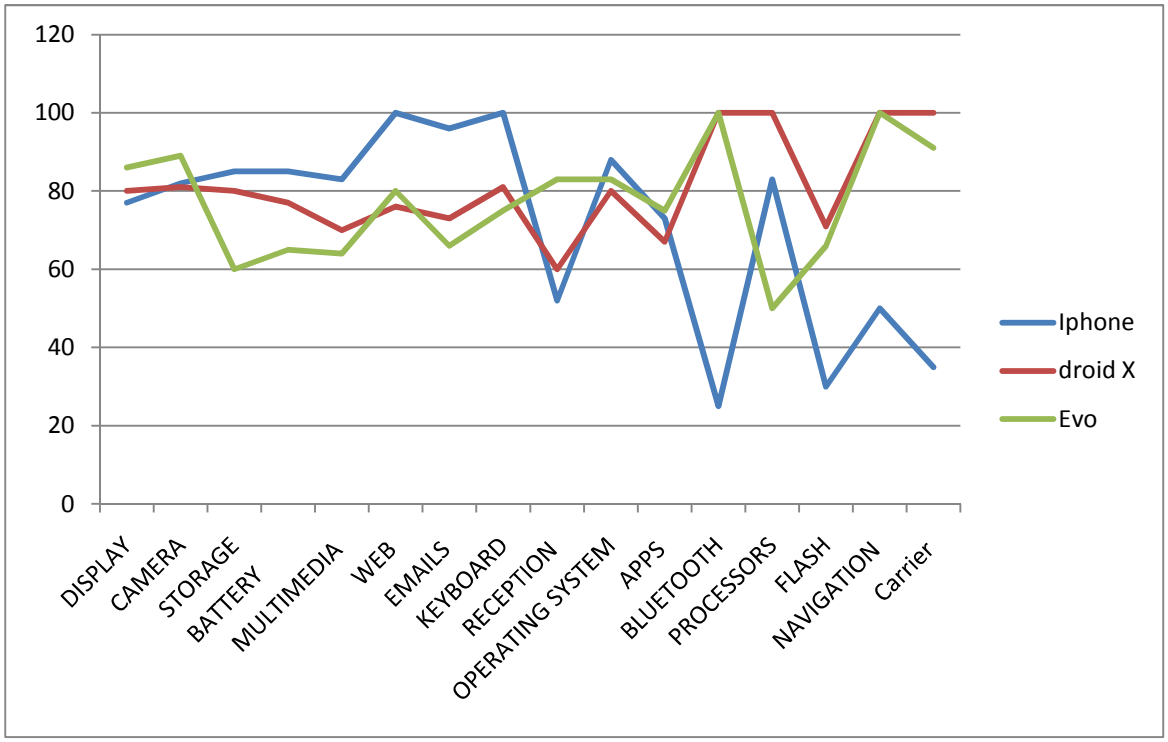


Figure 5.8 Comparison between smart phones with respect to features

The above diagram shows how different features of different products are compared. It clearly shows that the I-Phone though ranked as number 1 in both systems has very low value for Bluetooth, flash and regarding its carrier AT&T. If we have followed the regular approach and gave ranking based on the feature count we might have ended up in different rankings, but we properly gave scoring to each feature, that is the reason even though I-Phone has very low values for few features it is still number 1.

Best product for each feature, this can be found out by calculating positive percentage of overall occurrences.

For 3 products the best of each is given as below

Feature Name	I-Phone	Droid X	Evo
DISPLAY	77	80	86
CAMERA	82	81	89
STORAGE	85	80	60
BATTERY	85	77	65
MULTIMEDIA	83	70	64
WEB	100	76	80
EMAILS	96	73	66
KEYBOARD	100	81	75
RECEPTION	52	60	83
OPERATING SYSTEM	88	80	83
APPS	73	67	75
BLUETOOTH	25	100	100
PROCESSORS	83	100	50
FLASH	30	71	66
NAVIGATION	50	100	100
Carrier	35	100	91

Table 5.4 Table showing best product for each feature.

5.3 Evaluation Results

To evaluate the results we created instance files (arff files) as described in chapter 4. We use classifiers on these instance files to generate the results. The classifiers I used are Functional trees, random forest, SMO.

5.3.1 Evaluating Manual Generated with Manual Trained

The result of this manual generated which is created by reordering the original manually generated data, gives us a bench mark to compare the other results.

1. Functional Trees

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.714	0.005	0.909	0.714	0.8	0.93	2
	0.9	0.147	0.827	0.9	0.862	0.893	3
	0.853	0.096	0.897	0.853	0.874	0.892	4
Weighted Avg.	0.864	0.112	0.867	0.864	0.864	0.895	

Table 5.5 Evaluation Results for Manual data on manually trained data using FT

2. Random Forest:

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.143	0	1	0.143	0.25	0.963	2
	0.544	0.138	0.754	0.544	0.632	0.885	3
	0.922	0.433	0.676	0.922	0.78	0.885	4
Weighted Avg.	0.704	0.275	0.732	0.704	0.679	0.89	

Table 5.6 Evaluation Results for Manual data on manually trained data using Random Forest

3. Random Tree:

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.214	0	1	0.214	0.353	0.957	2
	0.567	0.112	0.797	0.567	0.662	0.868	3
	0.941	0.413	0.691	0.941	0.797	0.883	4
Weighted Avg.	0.728	0.254	0.758	0.728	0.708	0.882	

Table 5.7 Evaluation Results for Manual data on manually trained data using Random Tree

5.3.2 Evaluating the Automated Results with the Manual Results

These results will be main results to this thesis project. Here the training data is the manually annotated data and testing data is machine generated data using the methodology explained in chapter-3 and using the steps detailed in chapter-4.

1. Functional Tree

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.455	0.016	0.667	0.455	0.541	0.808	2
	0.638	0.432	0.539	0.638	0.584	0.582	3
	0.553	0.316	0.631	0.553	0.589	0.6	4
Weighted Avg.	0.584	0.348	0.593	0.584	0.584	0.605	

Table 5.8 Evaluation Results for Automated data on manually trained data using FT

2. Random Forest

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.091	0	1	0.091	0.167	0.885	2
	0.461	0.313	0.538	0.461	0.496	0.618	3
	0.741	0.494	0.594	0.741	0.66	0.653	4
Weighted Avg.	0.576	0.382	0.596	0.576	0.556	0.652	

Table 5.9 Evaluation Results for Automated data on manually trained data using Random Forest

3. SMO:

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0	0	0	0	0	0.5	2
	0.329	0.276	0.485	0.329	0.392	0.526	3
	0.794	0.609	0.56	0.794	0.657	0.592	4
Weighted Avg.	0.538	0.423	0.491	0.538	0.498	0.557	

Table 5.10 Evaluation Results for Automated data on manually trained data using SMO

5.3.3 Feature selection with automated data

This feature selection technique is used to select the most prominent features of a product, *i.e.*, features that are affecting the ranking of a product.

1. Feature selection with WrapperSubset evaluation with classifier Random Forest and selecting subset using greedy stepwise and finally verifying on functional trees classifier.

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.318	0	1	0.182	0.308	0.891	2
	0.48	0.37	0.507	0.48	0.493	0.555	3
	0.665	0.5	0.565	0.665	0.611	0.606	4
Weighted Avg.	0.594	0.361	0.621	0.592	0.605	0.638	

Table 5.11 Evaluation Results for Feature Selection for Automated data on manually trained data using Wrapper

And the top features selected as per this are quite match able to the one that we got it automatically and the ROC area is also 0.638, i.e, we got the result with quite a good accuracy.

Top features
Display
Storage
Multimedia
Bluetooth

Table 5.12 Table showing top features of the product, generated using wrapper.

5.3.4 Feature selection with Manual data

1. Feature selection with WrapperSubset evaluation with classifier SMO and selecting subset using BestFirst search method and finally verifying on RandomForest classifier.

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.455	0	1	0.455	0.625	0.817	2
	0.5	0.13	0.752	0.5	0.601	0.812	3
	0.912	0.494	0.643	0.912	0.754	0.824	4
Weighted Avg.	0.677	0.302	0.714	0.677	0.649	0.826	

Table 5.13 Evaluation Results for Feature Selection for Manual reordered data on manually trained data using Wrapper

Here in all the results we did not mention about the accuracy of the system, the reason for this is the we do not consider accuracy for these types of systems, *i.e.*, for systems of type which has more false positives or for system which has more empty fields. If we consider we will definitely get worst values, because the accuracy here gives the accuracy to find those false values than the truth ones.

Chapter 6 - Future Work and Limitations

In this thesis, I considered the problem of summarizing the reviews by considering the comparative sentences and give proper ranking to products by giving different weight age to each feature.

6.1 Principal Claims

1. Comparative sentences must be used in summarizing the product, or concluding any details of the product.
2. Feature selection cannot be general; it should be domain specific in order to get better results.
3. All features of a product cannot have equal value or weight age, each feature has their own importance and that importance should be rewarded.
4. Summarization or analysis should be from sentence level and feature level cannot be from top document level.
5. Providing alternatives to each product with respect to a feature.

After learning from Chapters 1 and 2, by implementing the methodology described in Chapter 3, with the experimental setup given in chapter 4 and by the results presented in Chapter 5. We are lead to several conclusions. In this chapter I present an interpretation of results and review all the claims.

The first goal of this thesis work is to use comparative sentences for better results, In chapter 3.2 I explained what is the use of using comparative sentences and the reason why by using the scoring of products alter by using comparative sentences.

The second goal feature selection cannot be general, *i.e.*, without using POS tags. The results chapter 5.1 clearly showed the features obtained using domain specific methodology. The features obtained using this strategy does not give all features and give unwanted features, which makes the task of comparing the products a bit tricky as there are no common set of features between products.

Chapter 5.2 clearly shows the effect of feature weight age and how it effects the ranking order of products. The results from chapter 5.2 clearly show the winner of three products

though one of the product has more negative features. The results also conclude that the feature weightage is most important factor and miscalculation of this feature score can change the results to maximum.

Here in this thesis the entire analysis of reviews is done in feature level, even the summarization part is done at review level and generalized to document level. The advantage of analyzing at feature level is we get detailed results, which is helpful for customers as well as product manufacturers.

Finally as claimed alternate to each product is what every customer is looking for, the results chapter 5.2 and 5.3 clearly shows and evaluates the alternate to each feature and top features of each product. Each result claimed is properly verified using various classifiers.

6.2 Limitations

The Limitations of this thesis work are

1. The data set that we used is less and has more noise in the dataset. With large data set we can generalize the things such as feature score and feature sets. The feature score and feature set that are calculated here are specific to this thesis. By training with large data set we might have overcome this problem.
2. The domain expert reviews are very less and became hard to find such expert reviews in order to verify the results claimed by me.
3. To determine the polarity I used a statistical approach based on Jane16 methodology, this approach uses database which is quite old and I improved the database to some extent but requires much improvement, because out of 77 occurrences of a particular feature the system is able to identify the polarity of just 55 or sometimes 35 instances.
4. This system can handle changes in data, but if we change the entire domain, then one need to specify the entire feature set calculate feature score. If there is no change in domain and just an addition of product, in that case a minor modifications to feature set makes this system automatic.

   For example: If a new smart phone came into market with an extraordinary feature, then that feature should be added manually to the corresponding feature section.

   The data should be crawled freshly whenever we introduce new product because the new product comparisons cannot be found in old database.

6.3 Future Work

Sentiment analysis is a new problem with improvement at every point; the main improvements for this thesis work are as below.

1. The main improvement can be the improvement of positive and negative schema used for finding the polarity. The schema should be improved with more database thought it is currently prepared with huge database, it still can be improved because the reviews are still pouring in the website. Moreover this database for determining the polarity works well for reviews or for experiments using customer reviews. but does not give good results with other type of experiments which needs to be improved.
2. This database which consist of positive schema and negative schemas, just let us know whether a corresponding feature is positive or negative. But it never tells us how much positive or how much bad a corresponding adjective is. This can be a good extension to this thesis work, finding how much good it is and how much bad it is instead of just finding it is good or bad. I have done the same thing for manual annotation which I did not and cannot extend to automatic but followed some statistical methods to determine the level of goodness.
3. Here I went back from domain independent strategy to domain dependent strategy, which may not be seems to be worth for few. But the domain independent strategy can also be improved by properly defining set of rules for tags while determining the feature.
4. This system can be improved for transferability: presently it is used for smart phone domains, and the system needs to be improved for transferability to other domains with less manual interactions. I tried to make as automated as possible for extensions to the same domain. But I found a lot of scope for improvement in the perspective of transferability.

Chapter 7 - Bibliography

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. *VLDB*, (pp. 487-499). chile.

best smart phones. (2010, Novemeber 23). Retrieved November 25, 2010, from cnet: http://reviews.cnet.com/best-smart phones/

Bing, L. (2006). *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data.* Springer.

books, w. (2010). *Data Mining Algorithms In R/Frequent Pattern Mining/The Apriori Algorithm*. Retrieved november 20, 2010, from wiki books: http://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_Apriori_Algorithm

Bruce, R., & Wiebe, J. (1999). Recognizing subjectivity: a case study in manual tagging.

Customers. (2010, October 12). *Cell phones*. Retrieved October 12, 2010, from Amazon: http://www.amazon.com/cell-phones-service-plans-accessories/b?ie=UTF8&node=301185

Customers. (2010, October 12). *Reviews*. Retrieved October 12, 2010, from cnet.

Dunne, K., Cunningham, P., & Azuaje, F. (2002). Solutions to Instability Problems with Sequential Wrapper-based Approaches to Feature Selection. *The Journal of Machine Learning Research.*

Esuli, A., & F, S. (2006). SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. *LREC: Language Resource and Evaluation (5th)*, (pp. 417-422).

Esuli, A., & Sebastiani, F. (2005). Determining the semantic orientation of terms through gloss classification. *CIKM*, (pp. 617-624).

Esuli, A., & Sebastiani, F. (2007). Pageranking wordnet synsets: An Application to Opinion Mining. *ACL: 45th Annual Meeting of the Association for Computational Linguistics.*

Fellbaum, C. (1998). *WordNet: an electronic lexical database,*. MIT Press.

Hatzivassiloglou, V., & McKeown, K. R. (1997). Predicting the Semantic Orientation of Adjectives. *35th Annual Meeting of the Association for Computational Linguistics*, (pp. 174-181). Madrid, ES.

Hatzivassiloglou, V., & Wiebe, J. (2000). Effects of Adjective Orientation and Gradability on Sentence Subjectivity. *COLING.*

Hsu, W. H., Welge, M., Redman, T., & Clutter, D. (2002). High-Performance Commercial Data Mining: A Multistrategy Machine Learning Application. *Data Mining and Knowledge Discovery,*, (pp. 361- 391).

Hu, M., & Liu, B. (2004). Mining and Summarizing Customer Reviews. *Knowledge Discovery and Data Mining Conference 04.*

I, W., & Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* San Francisco, CA: Morgan Kaufmann.

Jindal, N., & liu, b. (August 06-11, 2006). Identifying Comparative Sentences in Text Documents. *29th Annual International ACM SIGIR Conference on Research and development in information retrieval.* Seattle, Washington, USA.

Kamps, J., Marx, M. M., & Rijke, M. D. (2004). Using WordNet to Measure Semantic Orientations of Adjectives. *LREC-04, 4th International Conference on Language Resources and Evaluation*, (pp. 1115-1118).

Kondrak, M. A. (2008). A Comparison of Sentiment Analysis Techniques: Polarizing Movie Blogs.

li, w. (1999, january 1). *Zipf's Law*. Retrieved november 11, 2010, from http://linkage.rockefeller.edu/wli/zipf/index_ru.html

Liu, B. (2010). *Sentiment Analysis and Subjectivity.* Chicago: Department of Computer Science, University of Illinois at Chicago .

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating Classification and Association Rule Mining. *KDD.*

marianmedla. (2007). *Jane16*. Retrieved February 2009, from http://www.jane16.com/

Nation, P., & Waring, R. (1997). *Googlefight*. Retrieved October 13, 2010, from http://www.googlefight.com/

Newprosoft. (2009). Retrieved September 14, 2010, from http://www.newprosoft.com/web-content-extractor.htm#Demos

Pang, b., & Lillian, L. (2008). *Opinion Mining and Sentiment Analysis* (Vol. 2). Foundations and Trends in Information retrieval: Vol. 2: No 1-2, pp 1-135.

Pang, B., Lillian, L., & Shivakumar, V. (2002). Thumbs up? Sentiment classification using machine learning techniques. *EMNLP*, (pp. 79-86).

Prasov, Z. (2008). *Feature Subset Selection.* Michigan State University.

R, K., & G, J. (1997). Wrappers for Feature Subset Selection. *Articial Intelligence vol 97, NO 1-2*, (pp. 273 -324).

Sanmay, D. (2001). Filters, Wrappers and a Boosting-Based Hybrid for Feature Selection:. *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning,*, (pp. 74-81).

Smart phones Review. (2010). Retrieved October 22, 2010, from cell-phones.toptenreviews: http://cell-phones.toptenreviews.com/smart phones/

Tom, M. (1997). *Machine Learning.* McGraw Hill.

Turney, P. D. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02).* Philadelphia, Pennsylvania.

Turney, P., & Littman, L. (2003). Measuring Praise and Criticism: Inference of Semantic Orientation from Association. *ACM Transactions on Information Systems*, (pp. 315-346).

Waikato, M. L. (2008, November 1). *Attribute-Relation File Format*. Retrieved November 29, 2010, from Weka: University of Waikato: http://www.cs.waikato.ac.nz/~ml/weka/arff.html

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

Wiki. (2008, February). *Apriori algorithm*. Retrieved January 20, 2009, from Wikipedia: http://en.wikipedia.org/wiki/Apriori_algorithm

wikipedia. (2010, November 12). Retrieved November 18, 2010, from http://en.wikipedia.org/wiki/Tf%E2%80%93idf

Wikipedia. (2010, novemeber 10). Retrieved November 18, 2010, from http://en.wikipedia.org/wiki/Feature_selection

Wikipedia. (2009, January). *sentiment analysis: Definition of sentiment analysis*. Retrieved October 2010, from sensagent: http://dictionary.sensagent.com/sentiment+analysis/en-en/

Wilson, T., Wiebe, J., & P, H. (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing.*

Zhe, X. (2010). *A Sentiment Analysis Model Integrating Multiple Algorithms and Diverse Features.* Columbus, Ohio: Ohio state University.