CHALMERS

Social Media Data Mining and Inference system based on Sentiment Analysis

Master of Science Thesis in Applied Information Technology

ANA SUFIAN
RANJITH ANANTHARAMAN

Department of Applied Information Technology
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2011
Report No. 2011:073
ISSN: 1651-4769

Social Media Data Mining and Inference system based on Sentiment Analysis

Master's Thesis
Master of Science in Applied Information Technology

ANA SUFIAN
RANJITH ANANTHARAMAN

Department of Applied Information Technology
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2011

Social Media Data Mining and Inference system based on Sentiment Analysis
Master's Thesis

Master's Thesis 2011:073
Department of Applied Information Technology
CHALMERS UNIVERSITY OF TECHNOLOGY
SE – 412 96 Gothenburg
Sweden

Chalmers Reproservice
Gothenburg, 2011

Abstract

This thesis is about extraction of data from social networks and blogs and utilizing the captured data by processing it in order to make an analysis about certain specific products by using language processing. The data of four products namely iPhone, Sony Ericsson, Nokia and Samsung are collected from twitter and blogs.

The aim of this thesis to make a research in the field of natural language processing in order to find and implement an algorithm to solve the problem of measuring real-time comments made by user on products and produce inferences based on that. In collecting the data an own built web crawler was used. In continuation to that the blog entries were classified into relevant categories with an accuracy of 76.47% by using natural language processing algorithms where an investigation was made on developing a method to analyze negative, positive and neutral sentiments of tweets, comments and blog entries resulting in a satisfying 83.45% accuracy for an opinion mining application. Rates and scores were given to public opinions of the products which also were used to compare between the products where a statistical output was produced to show the results. The methods used in this project can be used for any specific product with public opinions.

Key Words: Natural Language Processing, Text categorization, Naïve Bayes, Term frequency- inverse document frequency, Sentiment Analysis, Opinion mining, Statistical Inference

Table of Contents

Preface

This work has been carried out at Department of Applied Information Technology, Chalmers University of Technology, Sweden during the period March to August of 2011. This work has been carried out under the supervision of Mr.Claes Strannegard, Assistant Professor at Department of Applied IT.

Acknowledgments

We would like to thank our supervisor Mr.Claes Strannegard for his support throughout the project. We would like to thank Mr.Nils Svangard for creating an opportunity for this project, for providing valuable suggestions on the field of work that could be researched on. We would like to thank Mr.Peter Ljunglof of Language Technology Group, Chalmers for his support and discussions on Sentiment Analysis and Classification problems. Finally we thank our friends in the rich internet community and forums whose valuable suggestions had been of immense use to us.

Notations

NLP	Natural Language Processing
TF-IDF	Term Frequency – Inverse Document Frequency

1. Introduction

Web data mining generally refers to crawling through the web locating and fetching from pages containing desired valuable information mostly with the use of web crawlers. Web crawlers can be built to fetch information of desired target or in other words they can be made application specific. They find high applications in search engines to give up-to-date information.

Nowadays social networks have covered hundreds of millions active and passive web users around the planet. The fast and exponential growth of social networking sites has proven undeniable, facilitating interconnection between users and high rate of information exchange. According to the Nielsen report in March 2009, Social Networking has been the global consumer phenomenon of 2008. Two-thirds of the world's internet population visits a social network or blogging site and the sector now accounts for almost 10% of all internet time **[1].** With this amount of large user information exchange, social media have become a good platform for research and data mining.

The valuable information retrieved from social networking sites can be utilized in many ways one of which can be to study, understand and predict the market for specific products which is very essential to improve qualities of the respective product. Due to scrutiny certain social networking sites are continuously updating their user-dependent privacy policies for their users, which in turn are becoming a bit of a challenge for mining them.

After collecting the desired information the most important part would be to understand the contents of this information. This where natural language processing comes into play. NLP is a field of computer science and linguistics concerned with the interactions between computers and human (natural) languages **[2].** One specific application in NLP that can be used for this purpose is sentiment analysis. It can be used to identify and extract subjective information from the information source collected**.** With all these processes and methods, it is possible to build a system which can extract application dependent information, process it and produce data which can be used for studying and deductions based on the information retrieved.

1.2 Motivation

With the rise of interconnectivity in our world with the different networks we have and with the amount of information shared, it is becoming highly important for harnessing this information on the web for various reasons. Based on the information collected applications such as market and stock predictions can be put into use. Especially this project focuses on its purpose in industries which are releasing their new products on the market will be eying on how the public responds in order to improve the relation between them and their customers. The data can be analyzed to study the nature of the market which then can be given as a feedback for the desired industry. One merit is that it can be done for any type of product as long as it is on the web.

It can open an area of research in solving the specified problem. There could be different approaches to it. The one and foremost most method which this project also happens to address is putting into use artificial intelligence and machine learning techniques. Sentiment analysis and different clustering and categorizing algorithms such as Bayesian methods and term frequency-inverse document frequency (tf-idf) methods are well established methods and widely used. The main purpose of this project is to collect data using an own-built web crawler, which focuses on the compromise between data quantity and quality, along with different APIs and also process the data from twitter and different blogs by the use of the above mentioned algorithms.

1.3 Limitations

During data collection one issue to be raised is making a compromise between large data extraction and low quality and lesser amount of data extracted with high quality. Blogs, forums and different social networks have different formats and outlines. Writing a specific program for general data collection is quite challenging. One of the limitations of this project was the need to cover most blogs on the net for collecting data of certain specific products there by collecting a larger amount of data. Due to the above mentioned problem certain unwanted noises are introduced.

Another limitation in this project was during the sentiment analysis phase where statements made by people are not always in correct grammar and with spelling errors. For tagging different parts of a sentence, the used sentence parser often finds wrong identifications of

sentence parts limiting the efficiency of the method used.

1.4 Related Work

This high rate of growth of information shared on the web has taken the attention of some researchers to use this information to analyze, predict situation based on the study. Some of the related researches are about analyzing the information for commercial purposes.

One study done by Bo Pang, Lillian Lee and Shivakumar Vaithyanathan **[3]**, focuses on classifying different movie reviews into three categories namely; positive, negative and neutral. They used different machine learning methods namely naïve bayesian method, support vector machines and maximum entropy method. In their research they collected a set of proposed negative and positive words for a movie review from an audience and used the three methods to classify whether a review is in one of the three categories. The reviews were taken from internet movie database (IMDb). With all the methods by changing the size of the word list they were able to achieve a peak accuracy of 82.9%.

The work done by Soo-Min Kim and Eduard Hovy in the paper "Determining the sentiment of opinions" **[21]**, is related to our thesis work in the context of grammatical parsing of sentences for sentiment detection. In their work they find the sentiments of individual entities or parts-of-speech in a sentence identify the regions between subject and the holder of that sentence and combine scores only within that region to arrive at the final sentiment score. We do proceed in a similar way during the sentiment analysis initial phase but do not follow the region based scoring method.

Several other attempts has been made on developing applications based upon sentiment analysis and can be found in the bibliography section. However we found many of the works to be focused on certain domains, few works are focused on relatively simple machine learning techniques which do not attempt to solve sentiment analysis to a depth, some work which are promising for grammatically correct sentences did not work on real time data. Hence we aim to develop a domain unrestricted product based sentiment analysis system that is based upon real time data.

2. Theory

The main driving idea behind the project is collecting data concerning different products of industries and analyzing the captured data to see how these specific products are performing on the market or "what do people actually talk about them?". It normally tries to answer "how much does a certain product score based on comments and texts from the social media?". Basically we aim to create a much generalized data retrieval engine and inference system that can infer aggregate opinions of the public on any product of interest. Products can belong to any domain say Electronics, Sports, Movies, etc.. Public opinions are mined from Social Network like Twitter, Blogs, and Forums. Inference we produce is based on the concept of Sentiment Analysis/Opinion Mining. Sentiment Analysis aims at making the system understand the Natural language expressed by people, fit a numerical score to the opinions in a range of positive/negative values. Feedback on a product given by actual users are very important than the information you can get from reviews/advertisements of the Company itself. We capture this piece of valuable information, process it and give you the results, seeing which you will be in a comfortable position in making further decision about the product.

2.1 Tools

The main tasks to accomplish in this project are

1. Collecting data from Twitter, Facebook, forums and Blogs
2. Analyzing the data using sentiment analysis and giving scores to each category.

For the above broad tasks, we have used different tools and performed different modules for each subtask to come under them. Mainly the programs are written and tested with java, with the help of html parsers for extracting data from pages and different API for mining and sentiment analysis accompanied with machine learning algorithms all to be discussed in sections 2.2 and 3. Following are the major tools used in our thesis source code implementation work.

I. **Twitter4j**: is an unofficial java library for twitter API which can be easily integrated with a java application with the twitter service by getting an authentication consumer key and consumer secrets. Within one run of the program it is possible to get a

.

maximum of 100 comments. It's also possible to search of comments with a set post date. In our project the search date is set to the current date with longitude and latitude location set to cover Sweden. For more information on Twitter4j library refer to **[10]**.

II. **Bing search java API 2.0**: is an API that can be integrated with a java application and used as a search engine. We used this API to retrieve the relevant blogs links of each specific product. A maximum of 30 links where visited for each run for each product. For more information on Bing Search API refer to **[8][9]**.

III. **Jericho HTML Parser 3.1**: is a java library allowing analysis and manipulation of parts of an HTML document, including server-side tags, while reproducing verbatim any unrecognized or invalid HTML. It also provides high-level HTML form manipulation functions **[4]**. Each text each relevant page of the blogs where extracted out by finding a specific pattern on how the text was arranged. For more information on Jericho parser refer to **[4]**.

IV. **Stanford Parser 5.18.2011**: is a sentence parser developed at Stanford University. It is a Java implementation of probabilistic natural language parsers, both highly optimized probabilistic context-free grammar (PCFG)(see foot note) and lexicalized dependency parsers, and a lexicalized PCFG parser. The original version of this parser was mainly written by Dan Klein, with support code and linguistic grammar development by Christopher Manning. For more information on the Stanford Parser refer to **[5]**.

V. **WordNet**: is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. We used this application for grouping blog contents which have similar message of words where the words are related based on this application or dictionary. For more information on WordNet refer **[6]**.

VI. **StopWordList1**: In computing, stop words are words which are filtered out prior to, or after, processing of natural language data (text). This stopword list is probably the most widely used stopword list. It covers a wide number of stopwords without getting too aggressive and including too many words which a user might search upon. This wordlist contains 429 words. For further information on StopWordList refer **[7]**.

3. Experiment

For sake of research and experimental purposes we collect data on four different products namely; iPhone, Nokia, Samsung and Sony Ericsson. All the data about each product are collected from twitter and blogs, methods for collecting data are discussed in following sections. The reason behind all the test products happen to be mobile products is because ample data is available on the same. In fact with the following proposed methods and techniques it's possible to evaluate any product's performance trends on the web. All the data are stored in a database for further use in the sentiment analysis phase.

On the collected data, we perform two levels of sentiment analysis, sentiment level and document level. As it should be obvious from the table, sentiment level analysis is carried out on the comments from blogs and tweets, and document level is based on contents of a website.

After performing the sentiment analysis we arrive at numerical scores for the above data and we need an inference system to infer conclusions from the experiments. Our Inference system uses simple statistics to make inferences. Inferences produced are given a product,

- What are the important features of a product and their respective numerical scores in the range -1 to +1. Negative score implies the particular feature is disliked by the public and positive score implies it is appreciated by the public. This inference is produced by two methods of sentiment analysis and it is only for research purposes.
- What is the best feature of a product
- What is the worst feature of a product
- What are the most spoken topics about a product
- Compare between two products and give their common features and scores. This can be of help in a way that you get to know the common aspects of two or more products and are provided with their respective scores, hence it helps you to make decisions among the products.

4. Proposed Algorithm

4.1 Data Collection

4.1.1 Web Crawler for blogs

We implement a web crawler to search the web in an automated manner and collect data of interest. Given a product, our desired data are the contents of any blog that writes about the product, comments made by users in those blogs or websites. To extract text from a HTML page we need a HTML parser. A parser is required in order to manipulate the HTML tags and retrieve the data present in tags of interest. In our case we used an open source Jericho HTML Parser.

To search the web for data, we need some seed list of websites to start with. For this purpose we use the Bing search API **[8][9]**, giving a simple search key to the Bing search engine, in our case the product name, we managed to take out the top links in the search results. We fixed the number of links to be fetched to top few based on two specific reasons:

1. Decrease in relevance
2. Decrease in popularity

As it is known with search engines, with increasing depth of list of links the lesser the content refers to the to our desired search query. In order to refer to relevant links we had to fix the number to links to check to 50. The same goes for popularity the most popular links do appear at the top most. The search results might contain results as different pages from the same domain and this might result in getting the crawler visit the same page more than once during depth crawling process. Hence for effective speed crawling purpose, we made the crawler take 50 seed url's of distinct domains.

Once we get the website's and blog's addresses in search result, for each of these URL's we crawl deeper inside each till a depth of 3 levels. Crawl depth is fixed as 3 after experimenting with depths from 1-5. Crawl depth is important when our crawler is trying to take data from too many websites which are not uniformly structured. Some websites might provide desired data in the index page i.e. first depth, but in some websites we might have to crawl deeper to retrieve the desired data.

Deciding on the data is desired or not is by itself another complex task which is not focused in much detail in our thesis. Noise is unwanted data that the web crawler collects. Noise in websites can be the text of advertisements, footer notes, etc. One of the attempts to reduce noise is verifying the data collected at various crawl depths and finding the crawl depth with least noise. Though there is no particular depth that can guarantee zero noise, considering the fact that we deal with quite a lot of data, we should compromise on noise allowing generalization of products and websites. Hence with factors discussed above, after experiments it was decided to fix the crawl depth as 3.

After reaching every HTML page by crawling through web, it is required to fix a format that crawler program should follow to take the data. If this is not done, then our program will take all data present in the page resulting in noise. Now it is extremely impossible to write a general rule for all pages in web, so that crawler will follow the rule and take data from all those pages. As it is known web pages in WWW are highly unstructured and is highly improbable that any two web pages follow the exact same structure.

4.1.2 Comment Extraction

With these challenges in place it is necessary to write rules that are as generic as possible, though we cannot make rules that are 100% generic. The common blogs on a product have a section ‘Comments’ under which users post their opinions. Some of the similar sections we identified are ‘Comments’, ‘Replies’. Since these are separate sections in a page they are placed in a header tag. We made a rule imposing the crawler to look up for data only under these specific header sections. Even if a web page does not follow the above assumed convention, crawler might only miss data from that page but not include any noise. Once crawler identifies these sections, we need a rule to make sure crawler program takes only the exact comment. On inspecting various blogs we came to a conclusion that comments are usually placed in a paragraph tag as: <*p*> comments of user </*p*>. All text within the paragraph tags are retrieved by the program until it finds another header section. A find of another header indicates the current comments section is ended.

The above rules does not guarantee a perfect information retrieval model but it does makes sense to assume web pages are ought to contain comments under ‘comments’ section and

comments are ought to be placed within the paragraph tags. To improve upon the information retrieval and to speed up the crawling process, we made the crawler program to spend time only on url's that contain '#comments' in it, thereby assuring the page contains comments.

4.1.3 Content Extraction

To extract the actual content written about a product by the author in a page, it is necessary to identify patterns that are common in most blogs where contents are posted. On inspection and verification of various sites we came to a conclusion to make the assumption that contents are placed within the divisional tags with their id's or class as 'Content', 'Entry', 'Post'. There could be more entities used in various other sites as there are no restrictions on their use, but to make a generalization that ensures data collection from most possible web pages with least possible noise is our aim in the current work. Using the above said assumption and with certainty that texts are places inside paragraph tags the crawler program could identify the contents of a blog. We also check if the retrieved content is visited by the crawler in its comment extraction module and if so the crawler skips the text as it is confirmed that the text belongs to a comment.

4.1.4 Twitter Data Collection

Twitter is a popular social media place to look for information about literally any product. Also the nature of tweets in twitter that its length cannot be greater tha 140 characters interests us because we deal with sentence level sentiment detection in the following sections. Twitter provides an open source library to access the tweets programmatically, there are many wrappers developed and in our case we use the twitter4j library. The API (application program interface) allows to input a search query along with various other preferences like

- Location of the tweet
- Language of the tweet, we are interested in only English tweets since we do not focus on detecting sentiments of other languages
- Username of the person who tweeted, but we do not take in this information.
- Tweets between certain period, or tweets since a date or tweets until a date. Twitter does not provide any data older than a week hence it is necessary to run the data retrieval program continuously in order to obtain updated tweets.

- Re-tweets, we can fetch the replies to a tweet also.
- Attitudes. Twitter has inbuilt program that classifies a tweet into positive, negative and neutral attitudes, pretty much the same we are attempting to do. But this feature was not made available in the twitter4j library we use and could be accessed only when searched manually through a browser.

With all the above features we can collect tweets of our choice. It was possible to collect the comments with all the features except attitude using our program. It was possible to collect 100 comments per run and managed to collect more comments in a real time basis, different amounts for each product.

Through the use of web crawler program and twitter data collector program we organize the collected data into database with respect to their products. Now that data is on hand we move to sentiment analysis module to associate the data with numerical score.

4.2 Sentence level Sentiment analysis

The aim of sentiment analysis is to detect the sentiment of a comment/sentence. A sentence can speak something positive, negative or can imply a neutral opinion about a feature. To be more exact, a sentence can imply positive opinion on some features, be neutral on some features and express a negative opinion on some features. So it is desirable to identify the features in a sentence, identify the opinions, and also identify which opinions are targeted to which of the features. Before going into the core pseudo code that generates the score we shall take a look at few concepts that are used in the algorithm.

4.2.1 Identify Opinions

What are opinion words? Opinion words are those words which express an opinion by itself, say amazing, wonderful, poor, bad etc. As a first step we frame a set of 10 positive opinion terms and 10 negative opinion terms as a seed list. For obvious reasons we need a better list of such words to cover varied sentences in English, hence we generate a bigger list in the following way.

For each word W_i in the Seed words list Add W_i to positive bag of words Add top 5 synonyms of W_i from WordNet to the positive bag Repeat above steps till the positive bag is of convincing size.

We follow the same algorithm to generate a negative bag of words. These set of words were then manually verified. In addition to make our opinion term list stronger we collect data from various other sources **[11][12][13]**. Finally we had around 357 words as positive terms and 527 words as negative terms.

4.2.2 Identify features

Features refer to the main topic on which the user aims to make an opinion on. It is vital to know the main topic of a sentence to infer the sentiment on such a topic rather than detecting sentiments of unimportant topics.

For example consider the sentence: "My brother had an Iphone which looks so cool".

In the above sentence brother and Iphone are topics the user is speaking about, but knowing how good or bad his brother is of no value to us, hence the important topic we wish to identify here is "Iphone".

The pseudocode for feature identification is as follows:

Pseudocode
For each word in a sentence, identify its Part-of-speech using Stanford POS Tagger**[14]** if the word is in Stopword list, skip the current word and proceed with the next word. Else If the word belongs to any of the following classes – NN(Noun Singular), NNS(Noun Plural),NNP(Proper Noun singular), NNPS(Proper Noun Plural) **[15]** Hit the Bing Search Engine with search key = current word + product name, record the search result size or the no: of hits. Else skip the current word and proceed with next word Rank the words according to the Bing hits and take top two ranked words as features of the sentence.

The above pseudo code infers desired words as features or subjects because of following reasons:

- In most sentences a user talks about a noun and we capture all forms of noun as shown in the pseudo code. Detecting the part-of-speech of a word is by itself a complex task and it is left to the Stanford POS Tagger library. There are few real time tweets/comments which fail using this library because they are not grammatically correct or constructed with mostly colloquial phrases.
- There might be more than two nouns in a sentence, and it is desired to find the nouns closely related with the product. This relation of nouns with product is inferred from the Bing hits which is logical.

4.2.3 Rules

Given the sentence we apply certain rules in the initial stage of sentiment analysis. These rules when applied produce a score on elements of the sentence. These rules are followed from a previous work on Sentiment detection **[16]**. These scores are used further in the algorithm.

Rule 1: Given a sentence assign +1 for positive opinion words, -1 for negative opinion words and 0 for context dependent words. Identifying opinion words are described earlier. Context dependent words are those related to the product in picture. For example, words like camera, battery, design, app, keypad, music etc. are context dependent words of any mobile phone.

Rule 2: This rule handles negation in a sentence. The word 'not' negates the meaning conveyed by the word succeeding it, and the word in succession is most probable to be an opinion word. In such a case it is important to handle negation of the opinion terms.
For example consider the sentence, "Samsung is not a good smart phone". The 'not' negates the opinion term 'good', 'good' has a score of +1 from rule1, and now after applying negation rule 'good' obtains a score of -1.

Rule 3: This rule handles but clauses in a sentence. Part of sentence preceding and succeeding the 'but' clause are usually oriented opposite in meaning to each other.
For example in the sentence "Samsung wave looks awesome but its price is costly", style is spoken good and price is spoken bad. Since we know 'awesome' is a positive opinion term we

can now infer that some negative is spoken about the context dependent word ‘price’.
Similarly is the case if we find an opinion term after the but clause.

Rule 4: This rule handles comparative sentences. Comparative sentences are those which uses ‘than’, ‘better than’, ‘higher than’ etc. to express positiveness about a feature and negativeness about another. Rule 4 works same way like Rule 3, if it finds an opinion preceding the comparator, it infers the contextual words succeeding the comparator as an inverse opinion.

4.2.4 Dependency

The information about dependency between words in a sentence is helpful in a way to know if the opinions are really intended to particular feature. We can use this information to score features more accurately. The Stanford dependencies provide a grammatical relation between words in a sentence **[17][18]** .

An example might make the concept more clear,

Sentence: “Iphone is cool but price is costly”

Output:

```
nsubj(cool-3, Iphone-1)
cop(cool-3, is-2)
cc(costly-7, but-4)
nsubj(costly-7, price-5)
cop(costly-7, is-6)
ccomp(cool-3, costly-7)
```

`nsubj(cool, Iphone)` means subject of the opinion term ‘`cool`’ is ‘`Iphone`’. In the following pseudo code we check if opinion terms and features of a sentence are dependent through the grammatical relations `nsubj` - ‘is subject of’ or `dobj` – ‘’is object of’.

4.2.5 Score generation

The pseudo code that associates a sentiment of a sentence through a numerical score is shown below. The following code identifies two best topics/features in a sentence, assesses the dependencies between features and opinion terms and scores the features in a range of real values between -1 to +1 based on a score-distance formula. All comments in the database are processed with this code and updated in the database. This processed data is later used by statistical programs to make inferences on the product.

Pseudo code

Given a Sentence S

- Identify opinion terms in S
- Identify top features f1, f2 in S
- Apply rules 1-4 on S and build the score table of Opinion words
- For each opinion term in score table,
- For each feature f,
- Check if the feature and opinion term are dependent, if yes proceed
- Score for f = $\frac{opinion\ term's\ score}{distance\ between\ opinion\ term\ and\ feature}$
- Check for emoticons,
- If positive emoticon is found add value to positive feature
- If negative emoticon is found reduce value from negative feature

We include distance as a factor in score calculation to detect how strongly in a sentence is an opinion expressed on a feature. It is most likely that opinions terms occur in a close range of the features, hence using the above formula we can avoid features getting scores from opinions which are not really intended.

Detecting emoticons is done at the final stage of scoring. Emoticons are a very straightforward way of expressing opinions. When we encounter a positive emoticon we add value to the positive feature and similarly on finding a negative emoticon we deduct value from a negative feature thus making it more negative.

Scores are normalized between -1 to +1 for clarity and prevent biasing over sentences of different length. When the score table is constructed we add scores of each opinion term in it using the formula: Score of Opinion term Oi = +1 or - 1 / no: of tokens in sentence
Final scores obtained for a sentence are not binary conclusions but can take any value in the range -1 to +1. This is much desired since one can the polarity of the sentiment i.e. knowledge that ‘how good or bad a feature is’ is more effective than the knowledge that ‘is the feature good or bad’. Also while dealing with large data, such real values can be of aneffective use in performing statistical inferences.

4.2.6 Secondary Score generation

The above algorithm does not guarantee a sentiment score for sentences of all kinds especially when dealing with real time sentences. The above algorithm works fine when the sentences are grammatically correct or at least near to correct. But people use many phrases, colloquial terms and don't attempt to strain to write grammatically correct sentences. The flaw in above algorithm occurs because of the dependency factor. The result is algorithm does not fit any score to the features in a sentence, and when many sentences in database are not scored it is a undesirable situation. Hence to overcome this problem we write a simple algorithm that does not take into consideration the grammatical structure of a sentence. The algorithm's scoring works in a simple manner as follows:

- Product name is assumed as a default feature
- Add Score = +1/no: of tokens for opinion terms identified as a positive opinion
- Add Score = -1/no: of tokens for opinion terms identified as a negative opinion
- Add Score = +1/no: of tokens to the feature if a positive emoticon is identified
- Add Score = -1/no: of tokens to the feature if a negative emoticon is identified
- Final score of feature = sum of scores of all opinion terms and emoticons

There are merits and de-merits in the current algorithm. De-merits are that it does not take into account the grammatical structure of a sentence, hence the score the algorithm produces does not guarantee its relevance to the features in a sentence. Accuracy is compromised to an extent here. On the other hand, it produces a tangible solution to the problem of no-score we had earlier and the loose scoring method does seem to be accurate enough practically is one of the observations we did. Though there is no support of theoretical proof the method seems to work well on most real time sentences which are grammatically poorly constructed. But we do not rely much on this algorithm primarily and we use this algorithm only on sentences which were not been able to be processed by our initial grammatical parsing sentiment score detection system.

4.3 Document level Sentiment analysis

The sentiment analysis was carried out in a document level also, i.e. detecting the sentiment of a whole document. A very controversial assumption was made in this module, we shall discuss about why the assumption is controversial. Assumption in carrying out sentiment analysis in document level is to

- Generate categories dynamically for documents from compiled contents
- Generate scores for comments given to the document i.e. perform sentiment analysis on sentence level and associate this result to the category of the document identified in previous step.

This second point told above forms the basis of assumption that all comments to a document is intended to the document only and does not discuss out of context of the document's content. For example, if the document or web page is a review about iPhone then we assume the comments found in the web page are related to iPhone only! This might be not be a very perfect assumption since people are not restricted to talk about the document topic only in their comments. But the assumption was made to suit different application. The document level sentiment analysis is simply a top-bottom form of sentence level sentiment analysis. Unlike the sentence level sentiment analysis, instead of finding the general opinion of a certain product, here we produce inferences of categories of products in order to understand how ones product is seen by the public opinion. For example, iPhone may be showing good results from the public opinion about its apps, but there could also be a possibility that there might be a bad result from the public opinion about its screen. So with this manner we can split the details and assess how a certain product is opinions in different categories. The document level sentiment analysis carried out in the way described above can serve these kinds of applications. Since we aim to research on the methods only we did not develop the application discussed now.

Following sections describe in detail the methods for text categorization of a web page's content and score generation for those categories.

4.3.1 Text categorization

The data retrieval programs have filled the database with contents of various blog's and website's along with the comments users have posted on them. Through this module we aim to categorize each blog into certain category and associate these categories with a sentiment score.

4.3.1.1 Naive Bayes method

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with naïve independence assumptions **[22]** in simple words takes an assumption that

the probability of one word appearing in a document doesn't affect the probability of another word appearing meaning they are totally independent. In order to work with naïve bayes there is a need for training data which are pre-classified. Based on the training data we have we can decide to which class our current document belongs to.

We have taken an assumption that with the stopwords thrown out of the document, the most frequent term in the document decides to which class that document belongs to. Therefore if one name of one specific class appears with its synonyms most frequently than the other sets of class name and its synonyms, then that document has a higher probability of being classified into that specific class. For each pre-classified class names, wordnet was used to produce a synonym set for calculating the probabilities. There is a certain bound b set where results below that lead to the classification of the document as neutral of not in any of the categories.

Pseudo code

Given a set of documents N and a document d to be classified;

- For each pre-classified document in N group each one of them with its class type and find the number of documents N_c in each class group.
- For each class c calculate the probability of the class $P_c = N_c/N$
- For each class c_j and for each term x_i in d, identify to which class d belongs to using the following formula

$$C = \max \left[Pc_j \prod P(^{x_i}/_{c_j}) \right]$$

The conditional probability $P(x_i/c_j) = {}^{n_{x_i,c_j}+1}\big/_{n_{c_j}+|v|}$

Where n_{x_i,c_j}= the number of times the term x_i appears in all the documents pre-classified as class c_j

n_{c_j}= the number of terms in all documents belonging to class c_j.

$|v|$= the number of vocabulary or synonym set of class c_j.

- If $C < b$, $C =$ neutral where $b =$ lower bound.
  Else $C = C$

With this classification procedure, a set of documents were classified and an accuracy of 72.22% was achieved.

4.3.1.2 Term Frequency-Inverse Document Frequency (tf-idf)

The tf-idf weight (term frequency–inverse document frequency) is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus **[19].** The weight of a word in a document can be calculated by taking into consideration the frequency of a word in the document under consideration and also its frequency in the entire document.

In this project this technique is used with an assumption that the most important word to the document can be taken as a class type for categorizing different documents. In order to minimize the error due to unwanted words being assigned with higher waits, in addition to the inverse document frequency analysis, stopwords where thrown out from the documents.

Pseudo code

For given set of documents D and each term in the document w

- For document d_i and for each term w_i in d_i calculate the frequency f_{w_i,d_i} of w_i in d_i and the frequency $f_{w_i,D}$ of $w_i$in all $d_i \in D$.
- The weight of w_i can be calculated as

  $$W_i = f_{w_i,d_i} * log\left({|D|}/{f_{w_i,D}} \right)$$

  Where $|D|$ id the number of terms in D
- Class C of d_i will be $max[W_i]$, where $w_i \in d_i$

By implementing the above procedure without the need for a training data set, its possible to categorize a set of documents into indefinite number of categories. Unlike the naïve bayes method we don't have a neutral category in this technique. With the above procedure we were able to classify 79.07% of sample documents correctly.

With both the above techniques we were able to classify documents. However both the algorithms had their own drawbacks to meet our purpose. In the case of the naïve bayes

method, even though we can classify the specific documents which are like the training set to be classified as one of the categories large portion of the collected data where classified as neutral meaning they are not like any of the chosen and pre-classified data we collected but of a different type and cannot be classified by this method. One question to raise here is why we had to fix the category types? The answer to that would be there are specifically important types of documents that can be analyzed and given as a feedback to the related user. Good example could be documents focusing on sales or customer service. With the specific pre-classified sales and customer service documents we can identify documents as such and can be given as a feedback to the responsible sales or customer service officer of that products company.

In regards to tf-idf it is possible to classify all documents and perhaps more easier to use in a way that there is no need for training data. However, for n set of documents it might be a case that we get n different classes. The number of categories is not pre-defined nor can be know before use.

To overcome the drawbacks we used a mixture of both the techniques where we classify the pre-specified categories as well as newly emerging categories and group identical ones together. With the use of this mixed technique we were able to classify all documents and leave out the least relevant ones which are not of any category. With this hybrid algorithm we were able to classify the documents with an accuracy of 76.4 %.

4.3.2 Score generation

The categories are associated with a sentiment score which is not done by using the content of blog directly, instead by performing the sentence level sentiment analysis on the comments posted by users on the same blog. This is quite unacceptable in the sense categories are generated out of contents, but scores are generated with the help of comments. But the method makes sense when it is in place to fit a confidence level for a web page. Scores from comments reveal the sentiment of people on data written on the contents of page. It makes relevance since public opinion need not be biased whereas the author of blog, company's blog may be biased to their own products.

The scores for comments are calculated the same way using sentence level sentiment analysis method discussed previously. The category of each blog receives its score as the sum of scores

of features found under the blog's comments. Thus a product has several blogs in our database and each blog is categorized into a category, each category has its own score. When a product inference is to be given we use this information about its categories and corresponding scores as the result.

5. Results and Discussion

5.1 Web Crawler

The data collected by Web crawler program and twitter crawler program are stored in the database, sample view of the collected data are shown in **Figures 1,2**.

COMMENT_ID	COMMENT_TEXT
nokia279	"I don't know about anyone else, but I bet that if WebOS was the OS that Nokia adopted their share price would have gone up, or stayed the same (assuming they also gave the message of "program in qt all 3 will be around for a while"). "
nokia667	pathetic
nokia281	It needs to be the complete package. On paper the N8 beat iPhone 4 in many aspects but not areas mainstream users are interested in (not helped by Nokia's poor marketing and Apple's incredible reality distortion field).
samsung134	The Case-Mate tough case is the perfect _Samsung Galaxy_ S ii case for those who want that extra bit of protection... http://dlvr.it/XVhY2
sonyericsson440	Sorry bb but I miss my sony ericsson
samsung205	stupid firefox "This app is incompatible with your AXIS Samsung GT-I5700." OTL
nokia99	Thats just cruel...:(
samsung565	samsung, you suck.
iphone884	You are an idiot.
nokia773	"I wanna f*ck nokia's ass. They lied, lied, lied, and lied.. to us."
iphone44	RT @K_Perezz: iPhone batteries suck!
nokia202	Not unlike your posts then.
iphone631	iPhone update is soooooooo slow
nokia935	Nokias stocks are not going up, #N9 not helping ...
nokia1002	@faenil: Nokias stocks are not going up, #N9 not helping ...
nokia256	Does anyone still play Angry Birds?

Figure 1: User comments from various blogs and website

COMMENT_ID	COMMENT_TEXT
136	Same feeling with rumors of iPhone 5 around the corner. RT @nicolepangLH: Dont announce the BB 9900 so soon. I butthurt with my Bold 3. :(
623	Wicked!...A Snap-On iPhone Lens That Shoots 360° Video | Co.Design http://t.co/CWdCoVj
97	@essbeevee I have not had a thing? Im on a crappy phone as lost my iphone...
496	the new iphone update is gonna be ridiculous
193	@Samtulp6_ @notcom because iPod Touch 2Gs and iPhone 3Gs dont require SHSH blobs
239	God I hate this twitter for iPhone app... Im about to delete n re-install Echofon.. With its sickening ass.
405	So happy that Ill be getting the iPhone... This blackberry can suck it! Lol
451	"@hausofmeena: @CrazyNerdyCool dont get the droid x! i hate it." GET THE IPHONE!
555	*wear. Im tired of this iPhone attempting to correct me. Its always wrong
733	@omgmus but spoil faster. K la I give in iphone gooooodddd laaa
379	@marcpaterson if you dont have an iPhone....dont matter
631	iPhone update is soooooooo slow
188	@BrandonRiddley yo emails dont come to your phone?! You obviously dont have an iPhone.

Figure 2: User tweets from Twitter

We collected the data for mainly 4 mobile brands and the **Table 1,2** illustrates the volume of data crawler has grabbed on the brands.

Table 1: Volume of data collected on each product

Product	Tweets from Twitter	Comments from Blogs	Total
Iphone	742	86	828
Samsung	600	95	695
Sony Ericsson	400	268	668
Nokia	794	880	1674

The following are the list of web pages the crawler has visited to take the data.

Table 2: Volume of web pages crawled on each product

Product	No: of Pages Crawled
Iphone	183
Samsung	33
Sony Ericsson	124
Nokia	72

5.2 Text Categorization

Given the data on a product, we dynamically generate categories that might be associated with dataset and rank them in order to arrive at best 5 categories. We associate every blog to a specific category using a combination of statistical Bayesian classifier and TF-IDF algorithm[19]. These best categories are in turn used in document level sentiment analysis explained in previous topics[**4.3**]. **Figures 3,4,5,6** illustrates the best categories generated for each product and the volume of blogs associated to those categories.

CATEGORY	COUNT(DISTINCTURL_ID)
apple	7
application	3
itunes	3
neutral	12
testing	3
tweak	6

Figure 3: iPhone

CATEGORY	COUNT(DISTINCTURL_ID)
battery	2
customer service	2
galaxy	5
glass	1
neutral	11
screen	2
technology	3

Figure 4: Samsung

CATEGORY	COUNT(DISTINCTURL_ID)
media	2
neutral	16
screen	1

Figure 5: Sony Ericsson

CATEGORY	COUNT(DISTINCTURL_ID)
neutral	28
phone	2
pixel	2
sales	3
symbian	10
windows	1

Figure 6: Nokia

Some of the results of text categorization are shown in **Figures 7,8,9,10**

CONTENT_TEXT	CATEGORY
"It is also important to note that if these glasses were made cheaper it would reduce the cost in manufacturing panels for these 3D TV&#65533;s. All this is being done by the SMPTE to make sure that this technology is affordable and the standardization efforts in 3D TV. It is also done so that both the industries continue making profit that is why it was decided that the Society of Motion Picture & Television engineers would be the one who would lead this standardization process. The work started by basically changing the format of Home Theatre making it suitable to 3D image viewing. In order for this to work the video content that was created, the AV equipment, viewing glasses required for the 3D TV&#65533;s everything was standardized to suit the Home Theatre. Then they took a leap forward by experimenting with 3D broadcasting in some corners of the world. The results were summed up and they showed a significant increase in the viewership.The word standardization means the process by which a business develops and agrees&#65533;s upon some technical standards. Since TV was not the place where the 3D was meant to be, they were certainly not the developers of this technology; there were some standards which had to be met. The Television companies couldn&#65533;t exploit this new technology to an extent that it causes friction between two of the biggest industries in the market which are the Cable TV and the Movies. Some rules had to be followed by which these companies can go about their business of using this new technology.While the 3D technology was being developed for movies, there were people working in television companies busy trying to develop newer forms of technologies like LCD, LED and Plasma, and then came a steep turn and an option whether to select a technology that was just being used in movies till today. That technology was 3D and the television industry sort of took a big risk by getting into this new field of electronics. Something that has never been tried before,"	technology

Figure 7: A blog's content and its classification into a category

CONTENT_TEXT	CATEGORY
Something a little more exciting is a new resolution category. Subscribe via RSS FeedNone of this may be real, but if it is, then it's up to Elop to SHOW and not TALK about changes coming to Nokia.What's interesting here are two things. Listed in the 360×640 category is something called the Nokia Oro. Unannounced T7-00 and X7-00 are also listed, as well as something called Nokia 601T/602T. In the VGA resolution, just above unannounced Nokia E6 is the Nokia 702T.That's of course if this is all true. This might be an aftermath of April 1 stories trickling down. It would be nice if the N8-01 is a successor to the imaging prowess of the N8-00, not just a N8 with more mass memory. Google search gives me random April 1 sources saying it's running Symbian^3, but with 1GHz processor.If you enjoyed this article, subscribe to receive more just like it.540×960. Still sticking with the 16:9 dimension. That's 1.5x the pixel width and height of the previous nHD displays, 518,400 pixels. nHD has 230,400 pixels, so 288,000 extra pixels. More than double, or 125 increase. Pretty much iPhone 4 territory (960×640) but due to screen ratio cuts off 100 pixels. The jump from nHD to qHD isn't as big as iPhone 3G only had 320×480. Then there's screen size to consider. Will we have 3.5″ still or make the move to 4-4.3″?We received this screenshot above from Janimatik. I'm not exactly sure where it's from and so whether to take any notice. Since it's an image, I can't pass this through a translator to have a guess what's being discussed (anybody here understand what's mentioned up top?)Tags: 960X540, featured, N8-01, Nokia About Jay Montano: Hey, thanks for reading my post. My name is Jay and I m a medical student at the University of Manchester. When I can, I blog here at mynokiablog.com and tweet now and again @jaymontano. We also have a twitter and facebook accounts @mynokiablog and facebook.com/MyNokiaBlog. Contact us at tips(@)mynokiablog.com or email me directly on jay[at]mynokiablog.com View author profile.We do know we are expecting some new hardware from Nokia as well as a revamped UI from Symbian. Some people may have had an advanced preview of this like the Product Manager from UK Mobile Network "Three" who said something amazing was coming to Symbian. Additional rumours point to an April 12 announcement. Is this possible? Elop has said that Nokia Products will ship closer to announcement. If true, we may have new devices on the market by May.DELIVER, DELIVER, DELIVER. Category: Nokia, Nseries Enter your email address below to receive updates each time we publish new content.	pixel

Figure 8: A blog's content and its classification into a category

CONTENT_TEXT	CATEGORY
int of the article was just to show how well HTC is doing, as opposed to how bad Nokia is (though how cool is it to beat the number one boxer if he's actually passed out on the floor and you're just kicking him?)I went on quite a departure there which I should give more explanation to but I'm rushing due to time constraints.Note, that we're still in further transition years. Two more apparently as we transition from Symbian to Windows Phone. Whilst all other manufacturers pretty much are just doing finishing touches to their "homes" , we're replacing the foundation, swapping walls, refitting the kitchen, replacing the stairs, taking out the living room, burning all the furniture.It's not the first time Nokia's taken such a big tumble however. Back in 2000, Nokia would have been worth, what, 222 Billion USD? I don't remember why it was that much at the turn of the year. From 1998, Nokia became the number 1 manufacturer of phones and to this day, has continued to be number 1. But stronger competition is nipping at its feet. For quite a while we've been waiting in the transition period. Hoping, longing that Nokia will finally have something to show us. 993 Billion TWD is 34.27 B USDNokia's Market cap dipped as low as 29.8B this year.During that time RIM(BlackBerry) had "eclipsed" Nokia. That also happened last year where RIM's market cap exceeded Nokia. This was during Nokia's low point. But it switches ever so quickly. RIM's market cap is now 29.1B (consistent over 31B for past few months but recent drop).HTC's operating margin, which measures the percentage of sales less the cost of making and selling phones, was 16 percent in the quarter ended Dec. 31. Nokia's was 7 percent over the periodCategory: RantBloomberg are reporting:I found the story after I read Andre's (@andref1989) tweet:HTC snapshot.http://www.bloomberg.com/news/2011-04-07/htc-surpasses-nokia-s-market-value-as-smartphones-drive-profit.htmlWhat you'll note is that the press will rarely if ever talk about Nokia's stocks rising, but love to talk about it dipping. Lowest point in 2010, Nokia's market cap was at 31B. For 5 months the value was consistent above 34B, pretty much 38B from September to January. On the memorable February 11 announcement, Nokia's stocks increased to 44.5 B amid speculations of something big happening on Capital Markets Day. It was an unbelievable announcement that not only would Nokia go Windows Phone, but Symbian (the world's largest smartphone OS) would be dropped.Ah, meh, have to go. Will edit this again perhaps.About Jay Montano: Hey, thanks for reading my post. My name is Jay and I m a medical student at the University of Manchester. When I can, I blog here at mynokiablog.com and tweet now and again @jaymontano. We also have a twitter and facebook accounts @mynokiablog and facebook.com/MyNokiaBlog. Contact us at tips(@)mynokiablog.com or email me directly on jay[at]mynokiablog.com View author profile.You do realise that market cap isn't actually a REAL value as such. If everyone decided to sell their shares, heck, even if 1/10th of people decided to sell their shares, you still think Apple is worth $311 billion? Enter your email address below to receive updates each time we publish new content.	sales

Figure 9: A blog's content and its classification into a category

CONTENT_TEXT	CATEGORY
"On the NC20: Yes, the screen might be bigger, but No this doesn't mean the stunningly good battery life of the NC10 has been thrown out of the window. Early tests show the NC20 gives 5.7 hours battery life in average conditions (average brightness with a web browser running, a document open, and wi-fi running).9. The Boatload of Extras8. Perfect for the home or officeXP is tried and tested, performs well even with low memory, and is compatible with the vast majority of software on planet earth.If you're trying to convince yourself the Samsung NC20's the right laptop, here's a decent start! 10 reasons we think the NC20 is a great buy.OK, so the battery lasts less than the NC10 battery did, but: 5.7 hours! That's almost as long as an iphone battery! (well, maybe not, but it's great anyway). The NC20's closest equivalent is the Dell Mini 12, which 'boasts' a measly 3 hours.Vista just doesn't work in a netbook. HP gave it a try with some of their Mini-notes.Any more reasons to buy the NC20? Or even not to buy the NC20? Let us know in the comments!The NC20 is primarily designed for mobile use. BUT it works just as well at home or in the office.A few years ago, you'd be lucky to get a 40GB hard drive in a laptop. Today you get 160 without compromising on weight or price. And, because this is a Samsung drive.It DOES takes extra space in your bag, and some will prefer to stick with the smaller screen of the older Samsung NC10. But if you want something to do anything other than surf the web or watch movies, a couple of extra inches goes a long way. If you're worried about size, the actual dimensions of the NC20 are 292mm x 217mm x 31mm.The Nano cruises where the old Intel Atom may have stumbled. It sits in the fast lane, cackling as its older half-cousin trundles along checking its mirrors. Lower heat consumption, faster performance and greater efficiency all help push up the battery life too.4. Microso"	battery

Figure 10: A blog's content and its classification into a category

In **Figure 10**, the blog talks about reasons to buy the product *NC20* essentially using its battery as a feature to boast upon. Though some other reasons were discussed crux of the content as we can see is *NC20's* powerful battery, and we get the output category the same ! Similar are the results of other blogs, some examples are shown in above figures 1,2,3.

5.3 Sentiment Analysis

The following is an example of detail steps involved in arriving at a sentiment score for a sentence. The sentence belongs to one of the real time comment found in our database. The algorithm for sentiment scoring is explained in the previous sections**[see 4.2]**.

Consider the input sentence/comment: " *my blackberry is better than my father's iphone*"
Initially score table is populated, score table contains the opinion terms, context dependent terms and their respective scores.
Initial Score Table: `{better=0.1111111111111111, iPhone=0.0}`
After applying Rules 1-4**[see 4.2.3]** the final score table is:
Final Score Table: `{better=0.25, iPhone=-0.1111111111111111}`
Next step is to find the subjects in a sentence.
As a first step, the Stanford POS Tagger initially identifies the parts-of-speech of individual tokens in a sentence, following is the output of the same:

```
my/PRP$ blackberry/NN is/VBZ better/JJR than/IN my/PRP$
father/NN 's/POS iPhone/NN :/: B/NNP
```

The POS Tags are standard abbreviations hosted in PennBank tree set**[20]**.

Tokens with POS tags representing all Noun-forms are taken as probable subjects. These subjects are concatenated with the product in context forms a search key. The search key is searched in Bing Search Engine through their Bing API for the number of hits. Higher the value of hits higher is the probability that the subject and product are related. Following is the result of this module:

```
{blackberry=72800000, iPhone=11300000, father=28300000}
```

The subjects are ranked based on their hits and best two subjects are chosen. In this case "iphone" is identified as a subject though it has fewer hits compared to "father", this is because the sentence belongs to the product "Iphone" and it is not logical to form a search key that contains two same words alone; "iphone Iphone" in this case. It would definitely result in fewer hits, hence we ignore such a result and simply choose such subjects. Result of the ranking module being:

```
subjects [blackberry, iphone]
```

Next process is to fit a score to the subjects identified. As explained in the scoring algorithm**[see 4.2.5]** it is important to take the score between a subject and an opinion term only if they are grammatically dependent on each other. To check the dependency between words in a sentence, we use the Stanford Dependency Tagger. Result of the dependency module is as follows:

```
poss(blackberry-2, my-1)
nsubj(better-4, blackberry-2)
cop(better-4, is-3)
poss(father-7, my-6)
poss(iphone-9, father-7)
prep_than(better-4, iphone-9)
dep(iphone-9, B-11)
```

In the above result you can see "`nsubj(better-4,blackberry-2)`", which says the subject of opinion term "better" is "blackberry". Score for subjects are calculated using the formula in**[see 4.2.5]**. Since "blackberry" is a subject of a positive opinion term it is ought to get a positive score. Iphone does not depend on any opinion term with a relationship "Subject", hence it retains its score obtained after applying the Rules1-4. Final result of the

sentiment detection of the sentence is as follows:

```
Blackberry = + 0.125
iPhone     = - 0.1111111111111111
```

The above is an example to show how a sentence is scored, all sentences in the database are subjected to sentiment analysis module thus producing two features per sentence and their respective scores. The **Figure 11** show a small part of the database containing sentences processed by sentiment analysis module.

COMMENT_ID	URL_ID	COMMENT_TEXT	PRODUCT_ID	SUBJECT1	SCORE1	SUBJECT2	SCORE2
nokia253	nokia14	I'm gettin sick of more and more angry birds series...bounce is better than angry birds.	3	nokia	-.06666666666666667	-	0
nokia200	nokia12	I think it is good price for it.	3	price nokia	.125	-	0
nokia134	nokia6	"My feedback is "its a beta, definitely not ready for release"."	3	nokia	-.09090909090909091	-	0
samsung19	samsung20	NX11 is a wonderful camera. Hope to see more lenses from both Samsung and third-parties!	4	camera	.25	Samsung	0
nokia468	nokia23	Remember not to take my words as promises of availability by Nokia. Software will be ready when it is ready...	3	Nokia Software	-.05	-	0
iphone527	iphone0	I recently gained 3 followers. I know this, thanks to @LazyUnfollow, this free IPhone app at http://bit.ly/ir5TZa	1	IPhone app	.058823529411764705	-	0
sonyericsson7	sonyericsson19	Excited ? can't thank you enough – what a joke – seems SE is excited about the business they lost over the past 6-7 months. Nice job SE, you've successfully ruined your market by joining hands with AT&T. way to go.	2	business	-.012195121951219513	SE	.004586199708150928
iphone830	iphone0	would have a cute iphone rather than sticking with this stupid N8	1	N8	-.08333333333333333	iphone	.08333333333333333
iphone829	iphone0	my blackberry is better than my fathers iphone!! :B	1	blackberry	.125	iphone	-.1111111111111
nokia4009	nokia0	you can give it a go, it worths, specially after mango update. go for nokia, they rule in building camera, everywhere :p	3	camera nokia	.045454545454545456	-	0

Figure 11: User Comments in database after Scoring

These scores as a standalone does imply something only on the sentence but not anything on the product in context. Hence we apply certain statistical measures on the processed sentences of a product to produce inferences!

5.4 Statistical Inferences

Feature Scores: The 5 best features of a product are presented to the end user as a result. It infers that for a product the system has found 5 top features which are mostly spoken about and their respective scores. These scores as explained previously are normalized and fall in the range -1 to +1. This eases a user to visualize the goodness or badness of a feature and also makes it easy to the user to compare the scores of different features. The feature

scores for a product are computed using both sentence level and document level.

Most Commented Topic: This statistic gives the idea on what are the most popular topics or features of a certain product. The result need not be necessarily a feature which is praised by many people; it can be a feature which is disliked by many people too! Nevertheless such features of a product are of interest to public since they can research on the reasons for its popularity. Alternate way is they can simply look at the next results of our system to check if the feature's popularity is a result of its goodness or badness.

Best Topic: The best feature of a product is given as a result to the user. The best feature implies it has the highest sentiment score among all other features of the product. Features with a very less frequency are not considered for this statistic thus ensuring the feature is valued high by a enough number of people comments.

Worst Topic: The worst topic result is the feature of a product that has obtained the minimum of scores among the features of a product.

The above results are compiled together for each of the 4 products and are shown in the **Figures 12,13.**

Figure 12: Results of iPhone and Nokia

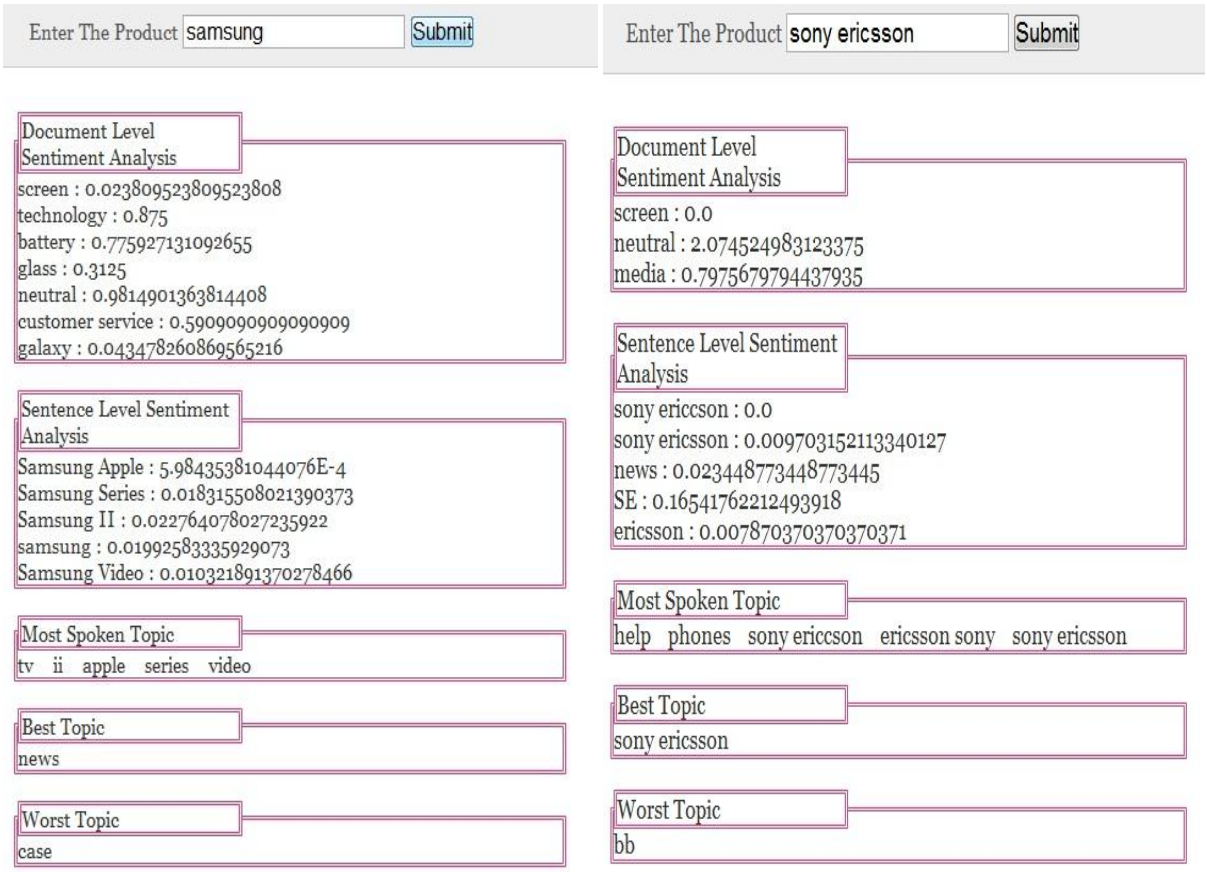

Figure 13: Results of Samsung and Sony Ericsson

5.4.1 Comparison between Products

Another useful statistic that could be taken out from the processed data set is the comparison of product features. Anyone interested in a product might wish to compare it with other products. We pick three categories namely “Price”, “Design” and “App”. The scores are taken only from the comments of each product those possess the category keywords as a subject. This ensures we take into account the very specific opinion on the feature and do not include any opinion casted on other features. Following **Figures 14,15,16** are the output of comparison module between different products.

Figure 14: Comparison Module Result (iPhone vs Nokia)

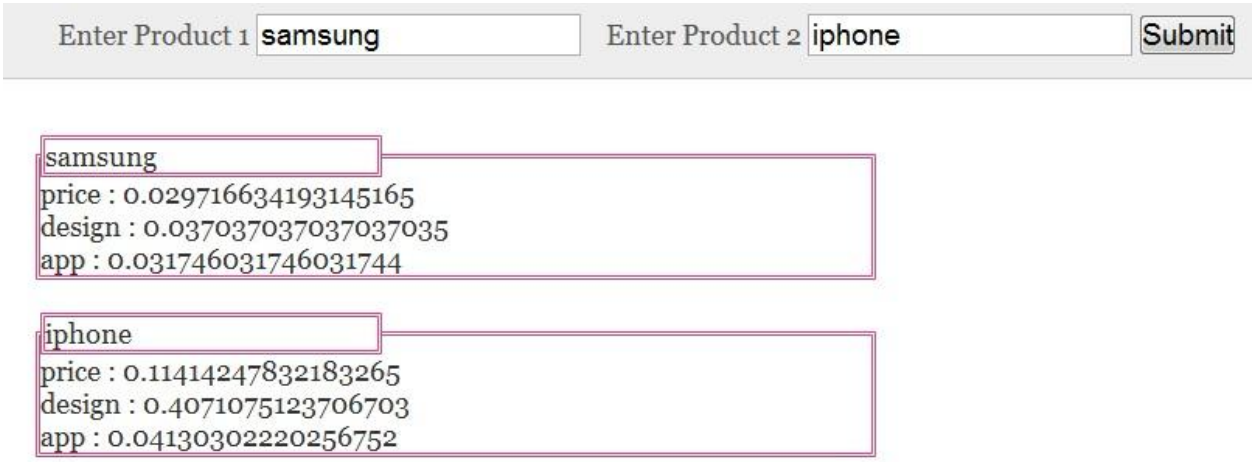

Figure 15: Comparison Module Result (Samsung vs iPhone)

Figure 16: Comparison Module Result (Nokia vs Samsung)

5.5 Analysis on limitations

Further we discuss few limitations of our analysis system. Consider the example in **Figure 17.** As seen in the 'COMMENT_TEXT' column, the user writes bad about "SE" in a sarcastic way, "*Nice job SE, you've successfully ruined your market.*.". The user has used positive words to tease SE the system (thanks, successful) and the system does not recognize this kind of sarcastic sentence constructions.

COMMENT_ID	COMMENT_TEXT	SUBJECT1	SCORE1	SUBJECT2	SCORE2
sonyericsson7	Excited ? can't thank you enough – what a joke – seems SE is excited about the business they lost over the past 6-7 months. Nice job SE, you've successfully ruined your market by joining hands with AT&T. way to go.	business	-.012195121951219513	SE	.004586199708150928

Figure 17: Sentence classified wrongly due to undetected 'Sarcasm'

There are words which behave ambiguously depending on the sentences where they are used. "Like" is a positive word when used in sentence as "*I like Nokia N8*". "*Like*" can be a neutral word in sentence like "*I would like to know if N8 has a 3G facility*". In the below **Figure 18,** sentences of comment id "*nokia490*" and "*samsung81*" are scored positive due to misinterpretation of "*like*" to be positive, whereas in fact "*like*" is only neutral in the two sentences. In the last sentence "*nokia145*" "*like*" is interpreted as desired and hence Nokia shop gets a positive score. It is difficult to arrive at a generalized rule that satisfies every possible sentence for any language. The work around for this problem can be to identify the part-of-speech of such ambiguous words in the sentence and make rules based on the clause in which the word is present in the sentence.

COMMENT_ID	COMMENT_TEXT	SUBJECT1	SCORE1	SUBJECT2	SCORE2
nokia490	I could never understand how something like this could have happened for such a European iconic firm…	nokia	.058823529411764705	-	0
samsung81	"oh and also, will we get a choice to upgrade the RAM like in the NC10? because 1 Gb of RAM these days isn't much."	RAM	.04	samsung	0
nokia145	Like Nokia clothing shop	Nokia shop	.3333333333333333	-	0

Figure 18: Example of wordd "like" taking different meanings

Next **Figure 19** shows the problem of noise extracted by our web crawler. The figure shows the content of a blog/web page extracted by the crawler program. The area highlighted in red is the undesired data or noise. It is undesired because it does not relate in any way to the blog contents and are advertisements or text from other sections of web page mistakenly extracted by the web crawler. This is due to the fact that data in web is highly unstructured and it is complex to design a noise resistant web crawler. The complexities involved in designing a web crawler are discussed in the initial sections **[see 4.1.1]**. Nevertheless the web crawler does not exclude any desired information but includes some amount of noise as shown. The easy solution for this problem is to crawl through websites of interest, limit the crawler to pages of particular structure. But we aim to create a system that is much generic and non biased towards any websites (during data collection stage), hence we had to ignore the noise present in the data.

CONTENT_ID	CONTENT_TEXT
nokia23	Enter your email address below to receive updates each time we publish new content.About Jay Montano: Hey, thanks for reading my post. My name is Jay and I m a medical student at the University of Manchester. When I can, I blog here at mynokiablog.com and tweet now and again @jaymontano. We also have a twitter and facebook accounts @mynokiablog and facebook.com/MyNokiaBlog. Contact us at tips(@)mynokiablog.com or email me directly on jay[at]mynokiablog.com View author profile. As I said there were about a development version of the Symbian Summer Update (which will NOT be named PR2.0). The name of the update ("Summer") suggests that the rollout of the new Symbian software is for this summer (around July) planned. Next Tuesday, this is officially announced and there will undoubtedly be more details released. If pn that day new phones will be presented is not yet known. In the rumor mill in recent weeks several new spotted Nokia Symbian devices, such as E6 and X7, but also on whether April 12 will see the light of day remains guesswork.If you enjoyed this article, subscribe to receive more just like it. Subscribe via RSS FeedSummer meaning possibly July apparently.Tags: Summer UpdateCategory: Nokia, SymbianYesterday, there as a bit of hubbub and much conversation about what Nokia could be announcing on April 12 in relation to what's supposedly going to be "New" with Symbian. New Software? New Hardware? Both?Last Monday, during the Tele Visie, I've had a chance to play with a demo version of the new Symbian version. The improvements in performance of the UI and browser, although this is a non-final version, were already noticeable. Photos I can not show you unfortunately.A lot of Symbian fans, current and past shared their frustrations and hope. 6 days to go, what could it be? BTW some Nokia folks are coming down to London with the WOMWorld folks for lunch next Tuesday which I may/may not be able to go to. I'll see my schedule:On April 12, I can guarantee you all 100 reporting, Nokia will announce the Symbian Summer Update. Yes, this is the 'big' update from Symbian ^ 3 which had already seen a few times in video demos include the C7 and E7. A few of the guys from Nokia will be coming over along with the WW/N team and a good spread for lunch "A post from Mobile-Cowboys (cheers spacemodel for the tip!) confirms the suspicion by some readers that the event is about the new Symbian Firmware Update. I don't know what the translator may have changed but apparently the PR2.0 update will NOT be called PR 2.0 but a Summer Update. Does this mean than the "Early 2011 50+ features" update is abandoned and is now merged with the new UI update, mentioned in Nokia Developer letter "First Major Update" for the Summer (and is now apparently called Summer Update). I did think that due to the lateness, they might as well just bundle all the updates together so it makes a bigger impact. For now, Opera Mobile 11 does a decent enough job taking over Mobile Browsing tasks."We haven't seen you in a while so we wondered if you were free for some lunch and a chat next Tuesday 12th at our offices from 12-2pm.

Figure 19: Extracted content of a blog. Noise is the area highlighted in red.

There are many examples of successfully handled sentences for sentiment detection as we had seen in the results section as well as certain scenarios which were not handled. We look at one more interesting scenario which was not handled and then move on to future work and conclusion. Consider the sentences in the **Figure 20.** Sentence with comment id "*nokia258*" has a positive score which is not the desired result. It is to be noted that these set of sentences are extracted from a blog and they belong to some discussion in particular.

Though the sentence is positive it is evident that the user agrees to one of the previous comments which are clearly negative. Usage of positive word "*thanks*" has resulted in the positive score, but the fact is that "*thanks*" was used to agree to negative opinions expressed by previous sentences on the product. The problem is we deal with sentiments casted by the sentence alone and do not overlook the sentence and its context within the discussion. Recording such knowledge about the context of discussion under which the sentence has come up can help in scoring the sentence to a more accurate level.

COMMENT_ID	URL_ID	COMMENT_TEXT	SUBJECT1	SCORE1	SUBJECT2	SCORE2
nokia252	nokia14	"I wish i was able to re-download my original Angry Birds and Seasons though; ovi claims i downloaded it too many times and wants me to pay for them again (i didn't though, contacted Nokia but they never replied)"	nokia	-.02564102564102564	-	0
nokia253	nokia14	I'm gettin sick of more and more angry birds series... bounce is better than angry birds.	nokia	-.06666666666666667	-	0
nokia254	nokia14	"Works here, though it lags a little bit ocassionally."	nokia	0	-	0
nokia255	nokia14	"I never understood the hype surrounding this game, it's rather boring."	nokia	-.09090909090909091	-	0
nokia256	nokia14	Does anyone still play Angry Birds?	nokia	-.16666666666666666	-	0
nokia257	nokia14	atleast the pigs are getting the rest	nokia	0	-	0
nokia258	nokia14	"Thanks, I did find it that way."	nokia	.14285714285714285	-	0

Figure 20: Series of comments from a blog. First 6 comments are negative while last comment misclassified as a positive since system has no knowledge of the situational context of the comment.

6. Conclusion

In this thesis work we developed a system that provides subtle inferences on a product based on public opinions. We wrote a web crawler to collect data, though it was not the prime goal of a thesis our crawler manages to retrieve data efficiently. We process the data using Natural Language Processing and Statistic based algorithms to make the system understand the opinions expressed by people and associate them with values. The system manages to process accurately the sentences which are and not grammatically correct. We dealt with feature extraction in a sentence, opinion words identification, grammatical parsing of the sentence to understand the relationship between features and opinion terms, score the features in a sentence and finally made some statistics to produce results to the user. We made the application run on a web environment where users can access the system through our website. The issues with our system being a web application are discussed in the following writing.

There was a trade off between Speed vs Accuracy we had to decide on right from the start of this project work. We chose to focus on Accuracy rather than speed since we thought there is a workaround for speed. There are two approaches we can see this work, one is to make the research work as a web application where people can use our program and see results from our website. Here speed is very important because any user would not wait for long time till our system can produce the result. Second approach is to send a results as a report to the user's mail box within half a day maximum. There is a need for time to produce GOOD Results because our system uses a web crawler that collects large amount of data from web, uses various English Language Parsers to analyze the gramatical structure of sentences etc. We can produce a score without these parsers but that is not an accurate inference, not close to the results we have now at all. Hence we resorted to produce better results compromising speed/time.

There were limitations in the system and were discussed in the end of previous section. On the merits side, we managed to develop a system that can infer about any product belonging to any domain; the scores generated by NLP module were good for the available data. And at last we can say that even though it might not only be these methods to use to solve the problems we investigated, our methods and algorithms have shown quite satisfactory results.

7. Future Work

There is a lot of scope for extension and future work on our thesis. The following are the possible areas we can work in future:

Improving the performance and quality of web crawler: Web crawler forms the basis of our thesis work since the entire module of sentiment detection depends on the data fed to it. One way to improve the performance of crawler is to maintain a record of websites crawled so that it does not re-visit the webpage again. Focusing on selected sites will improve the quality of data fetched which can be done if clients are interested in products of some specific domain. Higher the volume of data better is the accuracy of result. It is possible to write crawlers to fetch data from various other social media sites similar to twitter, e.g.: facebook, orkut etc thereby collecting more data!

Selection of NLP tools: We use the Stanford Parser to a large extent in our thesis. It is quite slow in processing sentences especially when sentences are long. Accuracy of the parser is not an issue since it recognizes well-formed sentences very precisely. Writing a own parser is not in our scope of future work, instead we can experiment on other language processing tools or updated versions of Stanford Parser.

Inclusion of context into our thesis framework might be very useful to analyze the sentences more accurately. Context means knowledge of sentences like if it is a question, if the sentence is a reply to previous statements etc can immensely help while declaring a subject as positive or negative.

Ambiguity resolution of certain words can be done to enhance the accuracy of sentiment scoring. The topic is discussed with an example in the earlier sections**[see 5.5]**.

Another interesting place we can extend our work is to extract the timeline of comments and record in the database. If the time of comment data is available we can infer about the opinion on the product in various periods of time. This area is known as trend analysis and is considered widely by organizations to know how their product’s popularity is trending from time to time.

References

[1] Global Faces and Networked Places, A Neilsen report on Social Networking's New Global Footprint, March 2009. Neilsen company.

[2] http://en.wikipedia.org/wiki/Natural_language_processing

[3] Thumbs up? Sentiment Classifcation using Machine Learning Techniques, Bo Pang and Lillian Lee, Department of Computer Science, Cornell University Ithaca, NY 14853 USA, Shivakumar Vaithyanathan ,IBM Almaden Research Center 650 Harry Rd. San Jose, CA 95120 USA

[4] http://jericho.htmlparser.net/docs/index.html

[5] http://nlp.stanford.edu/software/lex-parser.shtml

[6] http://wordnet.princeton.edu/

[7] http://www.lextek.com/manuals/onix/stopwords1.html

[8] http://msdn.microsoft.com/en-us/library/dd251056.aspx

[9] http://code.google.com/p/bing-search-java-sdk/

[10] http://twitter4j.org/en/index.html

[11] http://www.scribd.com/doc/4097251/A-to-Z-of-Positive-Words

[12] http://www.winspiration.co.uk/positive.htm

[13] http://eqi.org/fw_neg.htm

[14] http://nlp.stanford.edu/software/tagger.shtml

[15] http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

[16] Bing Liu. Sentiment Analysis and Subjecivity. Handbook of Natural Language Processing,2010

[17] Marie-Catherine de Marneffe and Christofer D.Manning. Stanford Typed Dependancies Manual, September 2008

[18] http://nlp.stanford.edu/software/stanford-dependencies.shtml

[19] http://en.wikipedia.org/wiki/Tf%E2%80%93idf

[20] http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

[21] Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. Proceedings of the 20'th International Conference on Computational Linguistics, 2004

[22] http://en.wikipedia.org/wiki/Naive_Bayes_classifier

Bibliography

[1] Christopher D.Manning, Prabhakar Raghavan, Hinrich Schutze. *Introduction to Information Retrieval. Cambridge University Press. May 2008*

[2] Peter Jackson and Isabelle Moulinier. *Natural Language Processing for Online Applications – Text retrieval, Extraction and categorization. John Benjamins Publishing company.*

[3] David Jensen and Jennifer Neville, *Data Mining in Social Networks. Knowledge Discovery Laboratory, University of Massachusetts.*

[4] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. 2004*

[5] Bo Pang and Lillian Lee. *Opinion Mining and Sentiment Analysis. Foundations and trends in Information Retrieval. 2008*

[6] Alec Go, L Huang, R Bhayani. *Twitter Sentiment Analysis. Association for Computational Linguistics. 2009*

[7] Tun Thura Thet, Jin-Cheon Na, Christopher S.G. Khoo,Subbaraj Shaktikumar. *Sentiment analysis of movie reviews on discussion boards using a linguistic approach. Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion. 2009*

[8] Carl W.Roberts. *A conceptual framework for Quantitative Text Analysis. Quality and Quantity. Springer – 2000*

[9] Theresa Wilson, Janyce Wiebe and Paul Hoffman. *Recognizing contextual polarity in phrase-level sentiment analysis. Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. 2005*

[10] Bing Liu, *Sentiment Analysis and Subjectivity. Handbook of Natural Language Processing, 2010*