To Implement

Opinion Mining and Sentiment Analysis

PROJECT

Prayas Suneja	Harvinder Singh Bhatia	Gagan Dwivedi
081235	081236	081242

Under the Supervision of **Mr. Ravikant Verma**

May 2012

Submitted in partial fulfilment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY

Department of Computer Science & Engineering

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

WAKNAGHAT

SOLAN, HIMACHAL PRADESH

Index

Chapter number	Topics			Page number
	Certificate			4
	Acknowledgement			5
	Summary			6
	List of Figures and Tables			7
1	**Introduction**			**8-17**
	1.1	Background and History		9
	1.2	Scope & Objective		10
	1.3	Project Summary		11
	1.4	Model of Opinion Mining		11
	1.5	Literature Review		12-17
		1.5.1	Opinion Mining-Bing liu	12
		1.5.2	Thumbs Up or Thumbs Down? –Peter D. Turney	14
		1.5.3	Mining the Peanut Gallery- Kushal, Steve, David	15
		1.5.4	Scary Films Good, Scary Flight bad- Scott Nowson	16
		1.5.5	Movie review mininig and summarization- Li Zuang, Feng Jing, Xiao Yan Zhu	17
2	**General Challenges**			**18-22**
	2.1	Contrasts with standard fact-based textual analysis		18
	2.2	Factors that make opinion mining difficult		19
3	**Program management**			**23-26**
	3.1	Software Development Model		23
	3.2	Software Architecture		24
	3.3	Software Design		25-26
		3.3.1	Use Case Diagram	25
		3.3.2	Data Flow Diagram	25
		3.3.3	Database Schema	26
	3.4	System Requirement		26
4	**Implementation and Schedule of Activities**			**27-28**
	4.1	Implementation		27
	4.2	Schedule of Activities		28

5	**WordNet**		**29-34**
6	**Result and Conclusion**		**35**
	6.1	Result	35
	6.2	Conclusion	36
	6.3	Future Work	36
7	**Code and Snapshots**		**37-53**
	7.1	Snapshots	37
	7.2	Code	39
8	**References**		**54**

CERTIFICATE

This is to certify that the work titled "Opinion Mining and Sentiment Analysis" submitted by Gagan Dwivedi, Harvinder Singh Bhatia and Prayas Suneja in partial fulfilment for the award of degree of B. Tech of Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor: ……

Name of Supervisor: Mr. Ravikant verma

Designation: Senior Lecturer.

Date:

ACKNOWLEDGEMENT

The project entitled as "**Opinion Mining and Sentiment Analysis"** which we develop refers to identifying and extract subjective information in source materials. A basic task in sentiment analysis is classifying the *polarity* of a given text at the document, sentence, or feature/aspect level — whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral.

Again, we would like to express our sincere thanks and gratitude to the project guide **'Mr. Ravikant Verma'** under whose guidance we are going to complete this project. He provides us the kind of strategies required for the completion of a task.

Signature of the students: ……

Name of Student: Gagan Dwivedi

Harvinder Singh Bhatia

Prayas Suneja

Date:

SUMMARY

Documentary information in the world can be broadly classified into two main categories, *facts* and *opinions*. Facts are object statements about entities and events in the world. Opinions are subjective statements that reflect people's sentiments or observations about the entities and events. . Much of the existing research on text information routing has been (almost exclusively) focused on mining and retrieval of factual information, e.g. information retrieval, web search and many other text mining and natural language processing tasks. Little work has been done on the processing of opinions until only in recent times. Yet, opinions are so significant that whenever one needs to make a decision one wants to hear others' opinions. This is not only true for individuals but also true for associations and organisations.

Searching opinion sources and monitoring them on the Web, however, can be a formidable task because a large number of diverse sources exist on the Web and each source also contains a huge volume of information. In many situations, opinions are hidden in long forum posts and blogs. It is very difficult for a human reader to search relevant sources, extract pertinent sentences, read them, summarize them and organize them into usable forms. **An automated opinion mining and summarization system is needed. *Opinion mining*, also known as *sentiment analysis*, grows out of this requirement.**

_____ _____

Signature of the Students

Name: Gagan Dwivedi

Harvinder Singh Bhatia

Prayas Suneja

Date:

Signature of Supervisor

Name: Mr. Ravikant Verma

Date:

List of Figures and Tables

Figure name	Figure description	Page number
Fig. 1	Feature based mining of a digital camera [1]	13
Fig. 2	Opinion comparison of two digital cameras [1]	13
Fig. 3	Proposed seed words[7]	19
Fig. 4	Software development model- Spiral model	23
Fig. 5	Software architecture- thin client model	24
Fig. 6	Use Case Diagram	25
Fig.7	Data flow Diagram	25
Fig. 8	Database schema	26
Fig. 9	WordNet design	31
Fig. 10	WordNet ouput	32
Fig. 11	Stemmer output	34
Fig. 12	Command line output	35
Fig. 13	Gui Screenshot- 1	37
Fig. 14	Gui Screenshot- 2	37
Fig. 15	Gui Screenshot- 3	38
Fig. 16	Gui Screenshot- 4	38

Table number	Table Description	Page number
Table no. 1	List of seed words[7]	27
Table no. 2	Project Schedule	28

1. Introduction

In common, opinions can be expressed on anything, e.g., a product, a service, an individual, an organization, or an event. The term *object* is used to indicate the entity that has been commented on. An object has a set of *components* (or *parts*) and as well set of *attributes*. Every component also have its sub-components and its set of attributes, and so on. Thus, the object can be hierarchically putrid based on the *part-of* relationship. Classifying evaluative texts at the document level or the sentence level does not notify what the opinion holder likes and dislikes. A positive document on an object doesn't mean that the opinion holder has positive opinions in all features of the object. Similarly, a negative document doesn't mean that the opinion holder dislikes everything about the object. In an evaluative document the opinion holder typically writes both positive and negative aspects of the object, although the common sentiment on the object could be positive or negative. To obtain such detailed aspects, going to the feature level is certainly needed. Based on this model presented earlier, below are the three key mining tasks are:

1. **Identifying *object features*:** For example, in the sentence "The picture quality of the camera is excellent ," the object feature is "picture quality". A supervised mock-up mining method is now proposed. An unsupervised scheme is also used. The technique basically detects frequent nouns and also noun phrases as features, which are usually genuine features. A number of information extraction techniques are also applicable, e.g., conditional random fields (CRF), hidden Markov models (HMM), and many others.
2. **Determining opinion orientations*:*** This task decides whether the opinions on the features given are positive, negative or neutral. In above sentence, the opinion on the feature "picture quality" is shown as positive. A number of approaches are now possible. A lexicon-based approach has been shown to perform fine .The lexicon-based approach basically uses opinion words and expressions in a sentence to determine the orientation of opinion on a feature. The reduction labeling based approach is given in .Various types of supervised learning are possible approaches as well.
3. **Grouping synonyms:** As the same object features can be expressed with dissimilar words or phrases, this task groups those synonyms collectively. Not much research has been done on the subject matter.

1.1. Background and History:

Documented information in the world can be broadly classified into two main category, facts and opinions. Facts are objective statements in relation to entities and events in the world. Opinions are subjective statements that echo people's sentiments or perceptions about entities and events. Much of the existing research on text information processing has been centred on mining and retrieval of factual information, e.g., information retrieval, Web search, and many other text mining and natural language processing tasks. Small work has been done in the processing of opinions until only recently. Opinions are so significant that whenever one needs to make a decision one wants to hear others' opinions. This is not only correct for individuals but also for organizations. One of the major reasons for the lack of study on opinions is there was little opinionated text before the World Wide Web. Before the Web, when an individual requires to make a decision, he typically asks for opinions from friends and families. When an organization wants to find opinions of the general public about its products and services, it usually conducts surveys and focused groups. With the Web, especially with the explosive development of the user generated content on the Web, the world has now changed. One can post reviews of products at merchant sites and convey views on almost anything on Internet forums, discussion groups, and blogs, which are collectively called the user generated content. Now if one wants to buy a product, it is no longer necessary to ask one's friends and families because there are ample of product reviews on the Web which give the opinions of the existing users of the product. For the companies, it may no longer need to conduct surveys, to organize focused groups or to employ external consultants in order to locate consumer opinions or sentiments about its products and those of its competitors. Finding opinion sources and monitoring them in the Web, however, can still be a formidable task because a large number of diverse sources exist on the Web and each source also contains a vast volume of information. In many cases, opinions are unseen on long forum posts and blogs. It is very difficult for a human reader to find appropriate sources, extract pertinent sentences, read them, summarize them and also organize them into usable forms. An programmed opinion mining and summarization system is needed. Opinion mining is also known as sentiment analysis, develops out of this need.

Although the area of sentiment analysis and opinion mining has recently enjoyed massive burst of research activity, there has been a steady undercurrent of interest for quite a while. One could count early on projects on expectations as forerunner of the area. Later work focused mostly on interpretation of metaphor, narrative, point of outlook, affect, evidentiality in text, and the related areas. The year 2001 or so seems to mark the beginning of widespread awareness of the research problems and opportunities that sentiment analysis and opinion mining lift and subsequently there have been literally hundreds of papers published in the subject.

Factors behind this “land rush” include:

• The increase of machine learning methods in the natural language processing and information retrieval;

• the availability of datasets for machine learning algorithms to be trained on, due to the blossoming of the World Wide Web and, specifically, growth of review-aggregation websites

• Realization of fascinating intellectual challenges and commercial and intelligence applications that the area proffers.

1.2. Scope & objective

An object *O* is represented with a fixed set of features, $F = f1, f2, \ldots, fn$, which includes the object itself. Every feature $fi \in F$ can be expressed with a finite set of phrases *Wi*, which are *synonyms*. That is, there is a set of corresponding synonym sets $W = W1, W2, \ldots, Wn$ for the *n* features. In evaluative document *d* which evaluates object *O*, an opinion holder *j* comments on a subset of the features $Sj \in F$. For each feature $fk \in Sj$ that opinion holder *j* comments on, he chooses a word or expression from *Wk* to describe the feature, and then expresses a positive, negative or neutral opinion on *fk*. The opinion mining task is to discover all these unknown pieces of information from the given evaluative document *d*.

Given an evaluative document *d*, mining outcome is a set of quadruples. Each quadruple is denoted by *H, O, f, SO*, where *H* is the opinion holder, *O* is the object, *f* is a feature of the object and *SO* is the semantic orientation of the opinion expressed on feature *f* in a sentence of *d*. Neutral opinions are ignored in an output as they are not always useful.

Given a collection of evaluative documents *D* containing opinions on an object, three important technical problems can be identified clearly there are more:

- *Problem* 1: Extracting object features that have been commented on in every document $d \in D$.
- *Problem* 2: Determining whether the opinions on the features there are positive, negative or neutral.
- *Problem* 3: Grouping synonyms of features as different opinion holders may use different words or expression to express the same feature.

1.3. Project Summary

Documented information in the world can be broadly classified into two categories, facts and opinions. Facts are purposeful statements about entities and events in this world. Opinions are subjective statements that reflect people's sentiments or observations about entities and events. Much of the existing research on this transcript information processing has (been almost exclusively) focused on mining and retrieval of factual information, for example. information retrieval, web search and many other text mining and natural language processing tasks. Small work has been done in the processing of opinions until only recently. Yet, opinions are so significant that whenever one needs to make a decision one wants to hear other people opinions. This is not only accurate for individuals but also accurate for the organizations.

Searching opinion sources and monitoring them in the Web, however, can be a formidable task because a large number of diverse sources exist on the Web and each source also contains a huge volume of information. In most of the cases, opinions are hidden on long forum posts and blogs. It is very not easy for a human reader to find relevant sources, extract pertinent sentences, read them, summarize them and organize them into usable forms. **An automated opinion mining and summarization system is needed. Opinion mining, which is also known as sentiment analysis, grows out of this requirement.**

1.4. Model of Opinion Mining

In general, opinions can be expressed on anything, for example, a product, a service, a topic, an individual, an organization, or an occurrence. The general term object is used to signify the entity that has commented on. An object has a set of components or elements and a set of attributes. Each module may also have its sub-components and its set of attributes. Thus, the object can be hierarchically mouldered based on the part of relationship.

Definition (object): An object O is an entity which can be a product, topic, persons, event, or organization. It is associated with a pair, O: T, A, where T is a hierarchy or nomenclature of components or parts and sub-components of O, and A is a set of attributes of O. Every component has its own set of sub components and attributes.

In this hierarchy , the root is objective itself. Each non-root node is a component or subcomponent of the object. Each linkage is a part of relationship. Each node is also related with a set of attributes. An opinion can be conveyed on any node and on any attribute of the node.

However, for an everyday user, it is too complex to use a hierarchical representation. To make it simple, the tree has been flattened. The word "features" is used to symbolize both components and the attributes. Using features for objects especially products is quite general in practice. Note that in this description the object itself is also a feature, which is the root of this tree. Let an evaluative document be d, which can be the product review, a discussion post or a blog that evaluates a particular object O. In the general case, d consists of the sequence of sentences d= ⟨s1, s2, …, sm⟩.

Definition (opinion passage on a feature): The opinion passage on the feature f of the object O evaluated in d is a group of consecutive sentences in d that conveyes a positive or negative opinion on f. This means that it is possible that a sequence of sentences at least one together communicates an opinion on an object or a feature of the object. It is also possible that a single sentence expresses opinions on more than one feature, for example, "The picture quality of this camera is good, but the battery life is too short".

Definition (opinion holder): The holder of a exacting opinion is a person or an organization that holds an opinion. In the case of product reviews, forum postings and blogs, opinion holders are always the authors of these posts. Opinion holders are important in news articles because they often unambiguously state the person or organization that holds this particular opinion. For example, the opinion holder in this sentence "John conveyed his disagreement on the treaty" is "John".

Definition semantic orientation of an opinion: The semantic orientation of opinion on a feature f states whether the opinion is positive, negative or impartial.

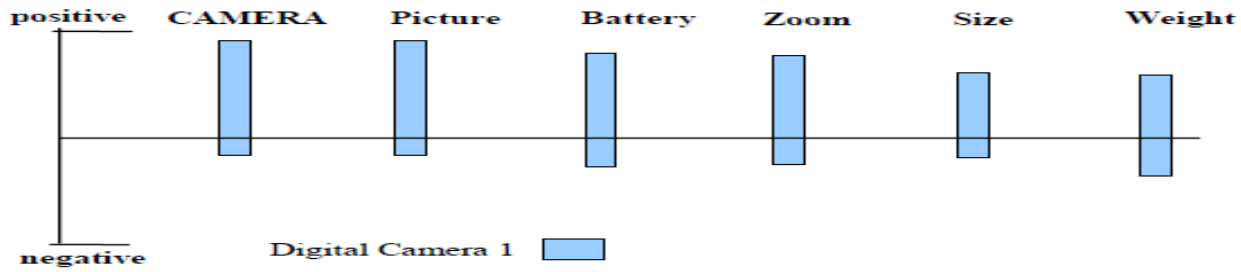
Putting things together, a representation for an object and a set of opinions on the features of the object can be defined, which is also called the feature-based opinion mining model.

1.5. Literature Review:

1.5.1. Research Paper on Opinion mining by Bing Liu[1]:

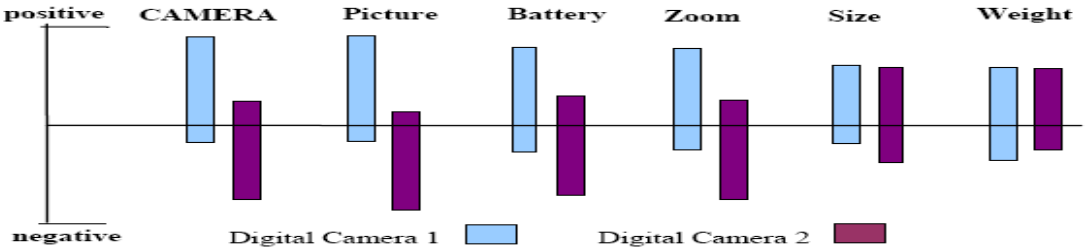
.

Given set of evaluative text documents D that contain opinions (or sentiments) about an object, the model aims to extract features and components of the object that have been commented on in each document **d ∈ D** and to determine whether the comments are positive or negative.



(A) Feature-based summary of opinions on a digital camera

Fig. 1



(B) Opinion comparison of two digital cameras

Fig. 2

Feature-Based Opinion Mining:

Classifying evaluative texts at the document level or the sentence level does not notify what the opinion holder like and dislike. A positive document on the object does not mean that the opinion holder has positive opinions on all portions or features of the object. Likewise, a negative document does not stand for that the opinion holder dislikes everything about this object. In an evaluative document e.g., a product review, the opinion holder typically writes both positive and negative features of the object, although the general emotion on the object may be positive or negative. To obtain such detailed aspects, going to the feature level is also needed. Based on the model presented earlier, three key mining assignments are there:

1. Identifying object features: For instance, in this sentence "The picture quality of this camera is astonishing," the object feature is "picture quality". A supervised pattern mining method is proposed. In an unsupervised method is also used. The technique basically finds common nouns and noun phrases as features, which are usually valid features. Clearly, many information extraction techniques are also relevant, e.g., conditional random fields CRF, hidden Markov models HMM, and many others.

2. Determining opinion orientations: This task decides whether opinions on the features are positive, negative or neutral. In the above sentence, the opinion on the attribute "picture quality" is positive. Again, many approaches are also promising. A lexicon-based approach has been shown to

execute quite well. The lexicon-based approach basically uses opinion words and phrases in a sentence to determine the orientation of an opinion on the feature. A relaxation labeling based approach is given in this document. Various types of supervised learning are potential approaches as well.

3. Grouping synonyms: As the same object features can be expressed with different expressions, this task groups those synonyms together. Much research has not been done on this topic. See for a challenge on this very problem.

1.5.2. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews by Peter D. Turney[2]:

Peter D. Turney presents easy unsupervised learning algorithm for classifying reviews as recommended (thumbs up) or not recommended (thumbs down). The classification of review is predicted by the average semantic orientation of the expressions in the review that contain adjectives or adverbs. A phrase has a positive semantic orientation when it has superior associations and negative semantic orientation when it has bad associations.

Shortcoming:

- average accuracy of 74%
- Difficult to classify reviews where the total is not necessarily the sum of parts. For example: average classification accuracy of movie reviews was 66% while that of automobile reviews was 80%.

This approach assumes that the feature sets produced will be too dissimilar. Therefore, in order to illustrate the results, a sample of words drawn from the Entertainment and Travel feature sets are shown in the table . The table contains illustrations of words that are significantly positive or negative in the either or both corpora. As one might expect, a number of words can be considered explicitly subjective, such as ‘luminous’ and ‘problem’. Differences in the use of such words suggest that merely being subjective word does not automatically suggest efficacy for sentiment classification. There are also number of topic-specific words which show an association with polarity despite being seemingly objective, for example ‘historic’, ‘song’ and ‘flight’. The effectiveness of the approach reported in this paper is further demonstrated by words which appear in both features set, yet indicate opposing polarities. In this entertainment corpora ‘book’ is used in a negative context, yet it is used positively in travel. These conflicting uses result in ‘book’ not appearing in the feature sets drawn from both subjects – its obvious usefulness for classification.

1.5.3. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews", by Kushal Dave, Steve Lawrence and David M. Pennock[3]:

Kushal Dave, Steve Lawrence and David M. Pennock have described different approaches and techniques for feature selection with quite varying accuracy. These approaches comprise Metadata and Statistical Substitution ,Linguistic Substitutions ,Language-based modifications and N-grams and proximity, Substrings

Limitations:

- Ambivalence and comparison
- Sparse Data
- Skewed Data

They are capable to obtain fairly good results for the review classification task through the choice of appropriate features and metrics, but identified number of issues that make this complexity difficult.

1. Rating inconsistency. Similar qualitative descriptions can also yield extremely diverse quantitative reviews from reviewers. In the most extreme case, reviewers do not understand the rating system and give a 1 instead of 5.

2. Ambivalence and the comparison. Some reviewers use terms that have negative connotations, but then write an equivocating ending sentence explaining that overall they were also satisfied. Others compare a negative occurrence with one product with a positive experience using the other. It is very difficult to split out the core assessment that should actually be correlated with the document's score. Mixed reviews introduce significant noise to difficulty of scoring words.

3. Sparse data. Many of the reviews are very short, and therefore we must be able to recognize a broad range of specific characteristics. Thresholding out the shorter reviews help out classification performance. Reviews from Amazon, when turned into a binary classification problem, are much easier to classify, at least in fraction because of their generally longer in size. In the C_ net corpus, more than two-third of words occurred in fewer than 3 documents.

4. Skewed distribution. On both sites, we discover that positive reviews were predominant, and certain products and product types have more reviews than the other. This is why, for example, the word "camera" is listed also as a top positive feature: this word appears in a large portion of the reviews, and most of those are positive. Although negative reviews were often longer than the positive ones, their language was often more varied, and achieving good recall on the negative set was not easy.

1.5.4. Scary Films Good, Scary Flights Bad by Scott Nowson[4]:

Scott Nowson describes preliminary work on feature selection for classification of review text by both sentiment rating and topics.

Limitation: The approach described works only on a specific type of data collected explicitly for purpose of classification, and hence is not suitable for a major fraction of opinionated data available ,e.g., texts must be labeled in some manner for sentiment or opinion texts must be from authors about whom demographic traits are known or can be determined.

One of most obvious conclusions one can draw from the two sets of experiments in this study is that topic is easier to classify than sentiment. It is perhaps more accurate, however,to say that despite perceived differences between texts at the extremes of sentiment, they are clearly less distinct than texts related to two of different topics. This is evidenced in the size of the feature sets. These were created to be the terms which distinguish between two or three corpora to a certain degree of significance. The number of strongly distinct terms is greater for the topic driven sets. A subsequent conclusion that this disparity in results suggests is that there is less variety in language used to convey the topic of a text than is used to express the sentiment therein. Pang and Lee's observation concerning the different levels to which different authors can express the same self perceived level of sentiment surely has no analogue in the topic. It is perhaps obvious to say, but measuring sentiment – as one does when asked to quantify with a rating – is entirely subjective; the fact that the sentiment is being expressed about movie or a hotel is entirely objective. As was intended, this study has shown the utility of the methodology employed here, a different form of divide and conquer. Employing feature selection to data stratified by topic has proven more effective than having all the data together. The approach to the subsequent feature selection has created reliable feature sets that lend themselves well to large scale of computing. The feature sets used here have been particularly small and, to varying degrees depending on the task, have performed well in some cases very well. Of course, there are natural criticisms that can be levelled at the very results. The most pertinent of these is that the results are too good, that they merely reflect over fitting and the feature sets will fail to generalise. The very nature of the approach is to select features best suited to the task in the hand. However, it is certainly true that for this preliminary study the features were created on the same data upon which they were then used to classify. Investigating beyond this specific subset of the collected data will be the first task following this study – not only drawing more data from the broader collection, but classifying on the entirely unseen data.

1.5.5. Movie Review Mining and Summarization by Li Zhuang, Feng Jing, XiaoYan Zhu[5]:

When a person writes a movie review, he probably comments not only movie elements e.g. screen-play, vision effects, music, but also movie-related people e.g. director, screenwriter, actor.For each feature class, if we remove the feature words with frequency lower than 1% of the total frequency of all feature words, the remaining words can still cover more than 90% of the feature occurrences. In movie reviews, some proper nouns, including movie names and people names can also be features. Moreover, a name may be expressed in the different forms, such as first name only, last name only, full name or abbreviation. The opinion words coming from statistical results on training data, the first 100 positive or negative words with highest frequency are selected as seed words and put to the final opinion keyword list. For each substantive in WordNet, we search it in WordNet for the synsets of its first two of the meanings. If one of the seed words is in the synsets, the substantive is added to the opinion word list, so that the list can deal with some unobserved words in the training data.

In case of feature-opinion pairs-A shortest path from the feature word to the opinion word is then detected. Then the part-of-speech and relation sequence of the path is also recorded.

2. General Challenges

2.1. Contrasts with standard fact-based textual analysis[6]:

The increasing interest in opinion mining and sentiment analysis is partly due to its potential applications, which we have just now discussed. Equally important are the new intellectual challenges that the field presents to the major research community. So what makes the treatment of evaluative text different from "classic" text mining and the fact-based analysis?

Take text categorization, for example. Traditionally, text categorization seeks to classify the documents by topic. There can be many possible categories, the definitions of which might be user- and application dependent; and for a given task, we might be dealing with as few as two classes binary classification or as many as thousands of classes e.g., classifying documents with respect to a complex taxonomy. In contrast, with sentiment classification we often have relatively few classes e.g., "positive" or "3 stars" that generalize across many domains and the users. In addition, while the different classes in topic-based categorization can be completely unrelated, the sentiment labels that are widely considered in previous work typically represent opposing if the task is binary classification or ordinal/numerical categories (if classification is according to a multi-point scale). In fact, the regression-like nature of strength of feeling, degree of positivity, and so on seems rather unique to the sentiment categorization (although one could argue that the same phenomenon exists with respect to topic-based relevance).

There are also many characteristics of answers to opinion-oriented questions that differ from these for fact-based questions As a result, opinion-oriented information extraction, as a way to approach opinion-oriented question answering, naturally differs from traditional information extraction i.e, IE . Interestingly, in a manner that is similar to the situation for the classes in sentiment-based classification, the templates for opinion-oriented IE also often generalize well across different domains, since we are interested in roughly the same set of fields for each of the opinion expression (e.g., holder, type, strength) regardless of the topic. In contrast, traditional IE templates can differ greatly from one of domain to the another — the typical template for recording information relevant to a natural disaster is very different from a typical template for storing bibliographic information.

These distinctions might make our problems appear deceptively simpler than their counterparts in fact based analysis, but this is far from truth. In the next section, we sample a few examples to show what makes these problems difficult compared to the traditional fact-based text analysis.

2.2. Factors that make opinion mining difficult[6]:

Let us begin with sentiment polarity text-classification example. Suppose we wish to classify an opinionated text as either positive or negative, according to overall sentiment expressed by the author within it.

Is this difficult task?

To answer this question, first consider the following example, consisting of only one sentence by Mark Twain: "Jane Austen's books madden me so that I can not conceal my frenzy from the reader". Just as the topic of the text segment can be identified by the phrase "Jane Austen", the presence of words like "madden" and "frenzy" suggests the negative sentiment. So one might think this is an easy task, and hypothesize that the polarity of opinions can generally be identified by set of keywords.

But, the results of an early study by the Pang et al. on movie reviews suggest that coming up with the right set of keywords might be less trivial than one might initially think off. The purpose of Pang et al.'s pilot study was to better understand the difficulty of the document-level sentiment-polarity classification and the problem.

Two human subjects were asked to pick keywords that they would consider to be good indicators of positive and negative sentiments. The use of the subjects' lists of keywords achieves about 60% accuracy when employed within straightforward classification policy. In contrast, word lists of the same size but chosen based on examination of corpus' statistics achieves almost 70% accuracy — even though some of the terms, such as "still", might not look that intuitive at first.

	Proposed word lists	Accuracy	Ties
Human 1	positive: *dazzling, brilliant, phenomenal, excellent, fantastic* negative: *suck, terrible, awful, unwatchable, hideous*	58%	75%
Human 2	positive: *gripping, mesmerizing, riveting, spectacular, cool, awesome, thrilling, badass, excellent, moving, exciting* negative: *bad, cliched, sucks, boring, stupid, slow*	64%	39%
Statistics-based	positive: *love, wonderful, best, great, superb, still, beautiful* negative: *bad, worst, stupid, waste, boring, ?, !*	69%	16%

Fig. 3

However, the fact that it may be non-trivial for humans to come up with the best set of keywords does not in itself imply that the problem is harder than the topic-based categorization. While the feature "still" might not be likely for any human to propose from introspection, given training data,

its correlation with the positive class can be discovered via a data-driven approach, and its utility at least in the movie review domain does make sense in retrospect. Indeed, applying machine learning techniques based on unigram models can achieve over 80% in accuracy , which is much better than the performance based on the hand-picked keywords reported as above. However, this level of accuracy is not quite on par with performance one would expect in typical topic-based binary classification.

Why does this problem appear harder than the traditional task when two classes we are considering here are so different from each other? Our discussion of algorithms for classification and extraction will provide a more in-depth answer to this question, but the following are a few examples showing that the upper bound on problem difficulty, from the viewpoint of machines, is quite high. Note that not all of the issues these examples raise have been fully addressed in this existing body of work in this area.

Compared to topic, sentiment can often be expressed in a more subtle manner, making it difficult to be identified by any of sentence or document's terms when considered in isolation. Consider the following as examples:

• "If you are reading this because it is your darling fragrance, please wear it at home exclusively, and tape the windows shut." review by Luca Turin and Tania Sanchez of the Givenchy perfume Amarige, in Perfumes: The Guide, Viking 2008. No ostensibly negative words thus occur.

• "She runs the gamut of emotions from A to B." Dorothy Parker, speaking about Katharine Hepburn.

No ostensibly negative words then occur.

In fact, the example that opens this section, which was taken from the following quote from Mark Twain, is followed by a sentence with no ostensibly negative words:

Jane Austen's books madden me so that I can't conceal my frenzy from the reader. Everytime I read 'Pride and Prejudice' I want to dig her up and also beat her over the skull with her own shin-bone.

A related observation is that although the second sentence indicates an extremely strong opinion, it is difficult to associate the presence of this strong opinion with specific keywords or phrases in this very sentence.

Indeed, subjectivity detection can be a difficult task in itself. Consider the following quote from Charlotte Bront¨e, in letter to George Lewes:

You say I must familiarise my mind with this fact that "Miss Austen is not a poetess, has no 'sentiment' " you scornfully enclose the word in inverted commas, "has no eloquence, none of ravishing enthusiasm of poetry"; and then you add, I must "learn to acknowledge

her as one of the greatest artists, of the greatest painters of human character, and one of the writers with the nicest sense of means to end that ever lived".

Note the fine line between facts and the opinions: while "Miss Austen is not a poetess" can be considered to be a fact, "none of the ravishing enthusiasm of poetry" should probably be considered as an opinion, even though these two phrases (arguably) convey similar information. Thus, not only can we not easily identify simple keywords for subjectivity, but we also find that patterns like "the fact that" do not necessarily guarantee the objective truth of what follows them — and bigrams like "no sentiment" apparently do not guarantee absence of opinions, either. We can also get a glimpse of how opinion-oriented information extraction can be a difficult task. For instance, it is non-trivial to recognize the opinion holders. In the example quoted above, the opinion is not that of the author, but the opinion of "You", which refers to George Lewes in the particular letter. Also, observe that given the context "you scornfully enclose the word in inverted commas", together with the reported endorsement of Austen as a great artist, it is clear that "has no sentiment" is not meant to be a show-stopping criticism of Austen from Lewes, and Bront¨e's disagreement with him on the subject is also subtly revealed. In general, sentiment and subjectivity are quite context-sensitive, and, at a coarser granularity, quite domain dependent in spite of the fact that the general notion of positive and negative opinions is fairly consistent across different domains. Note that although domain dependency is in part a consequence of changes in vocabulary, even the exact same expression can indicate different sentiment in the different domains.

For example, "go read the book" most likely indicates positive sentiment for book reviews, but negative sentiment for the movie reviews. This example was furnished to us by Bob Bland. We will discuss topic sentiment interaction in more detail in It as does not take a seasoned writer or a professional journalist to produce texts that are difficult for the machines to analyze. The writings of Web users can be just challenging, if not as subtle, in their own way — see for an example. In the case of it should be pointed out that it might be more useful to learn to recognize the quality of review

Still, it is interesting to observe the importance of the modeling discourse structure. While the overall topic of a document should be what the majority of the content is focusing on regardless of the order in which potentially different subjects are presented, for opinions, the order in which different opinions are presented can result in completely opposite overall sentiment polarity.

In fact, somewhat in contrast with topic-based text categorization, order effects can completely overwhelm frequency effects. Consider the following excerpt, again from a movie review:

This film should be the brilliant one . It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a great performance.

However, it can not hold up. As indicated by the inserted emphasis, words that are positive in orientation dominate this excerpt, and yet the overall sentiment is negative because of the crucial last sentence; whereas in traditional text classification, if a document mentions "cars" relatively frequently, then the document is most likely at least somewhat related to the cars.

Order dependence also manifests itself at more fine grained levels of analysis: "A is better than B" conveys the exact opposite opinion from the "B is better than A". In general, modeling sequential information and discourse structure seems more crucial in the sentiment analysis.

As noted earlier, not all of the issues we have just discussed have been fully addressed in literature. This is perhaps part of the charm of the emerging area. In the following chapters, we aim to give an overview of a selection of past heroic efforts to address some of these issues, and march through the positives and the negatives, charged with unbiased feeling, armed with the hard facts.

3. Program management

3.1. Software Development Model:

- The model which is being used in this project development is ***Spiral model*** as it combines elements of both designs and prototyping-in stages.
- Process is represented as a spiral rather than as sequence of activities with backtracking.
- Each loop in the spiral represents phase in the process.
- Also known as the spiral lifecycle model, it is the systems development method (SDM) used generally in information technology(IT).

The Spiral Model:

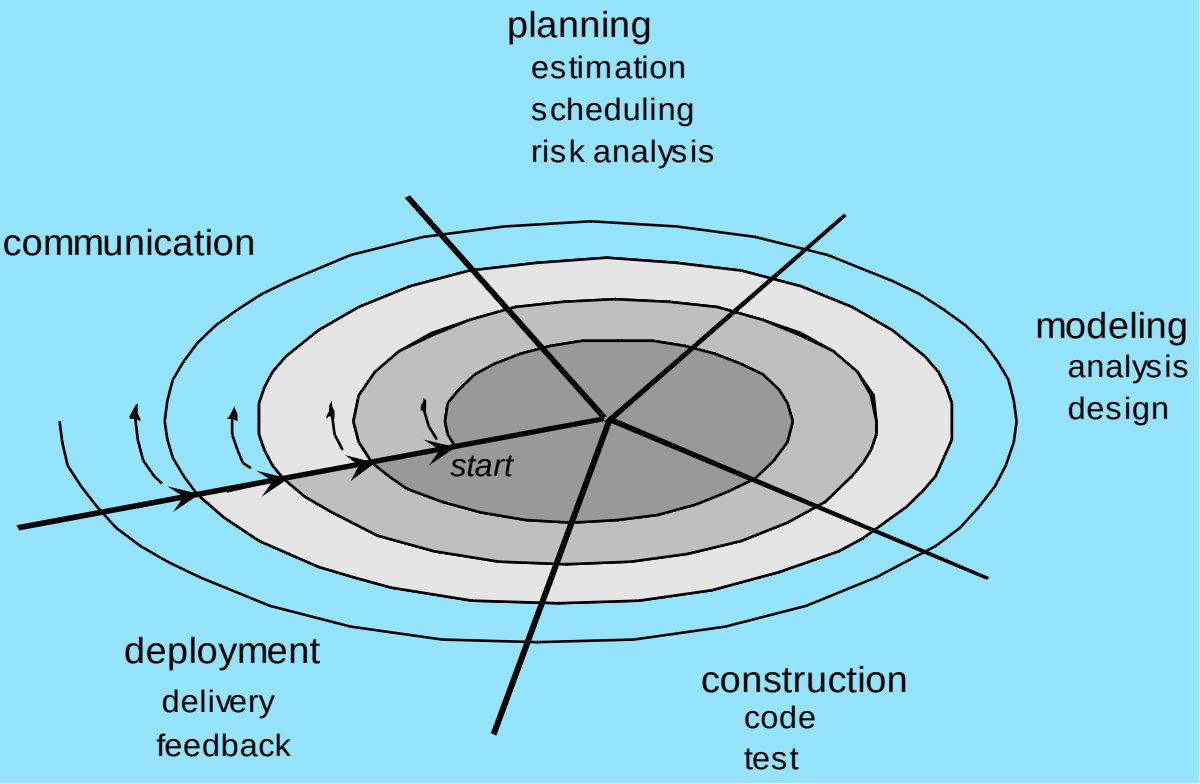


Fig. 4

Spiral Model Sectors:

- Objective setting
  - Specific objectives for a phase are also identified.
  - Risks are assessed and activities put in the place to reduce the key risks.
- Development and Testing
  - The next phase of the project is developed and then tested.
- Deployment
  - The changes are deployed, and results are then analysed.

- Planning
    - The project is reviewed and the next phase of the spiral is then planned.

Advantages:

- The spiral model is a realistic approach to the development of large-scale systems and the software.
- Spiral model demands a direct consideration of technical risks at all stages of the project, and, if properly applied, should reduce risks before they could become problematic.
- This model also combine the advantages of top-down and bottom-up concepts.

Disadvantages: Spiral model demands considerable risk assessment expertise and relies on this expertise for the success.

3.2. Software architecture

The Architecture of our project is the ***Client-Server with Thin Client Model.***

It is a distributed system model which shows how data and processing is distributed across range of components.

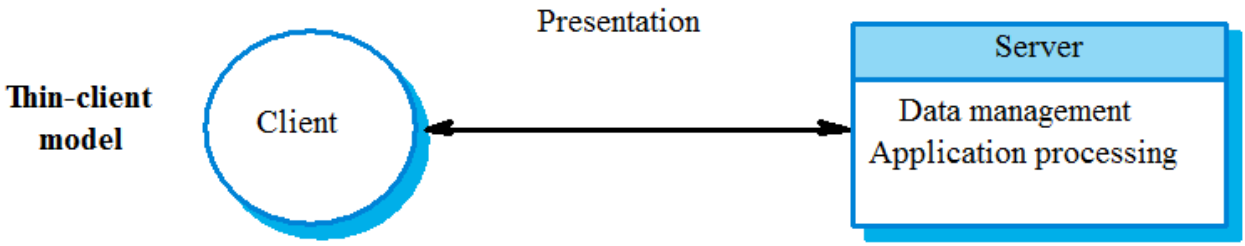


Fig. 5

It contains:-

- Set of stand-alone servers which provide specific services such as printing and data management.
- Set of clients which call on the services.
- Network which allows clients to access the servers.

The application is modelled as a set of services that are provided by servers and a set of clients that use the services.

Clients know of servers but servers need not know of the clients.

Clients and servers are the logical processes

The mapping of the processors to processes is not necessarily 1 : 1.

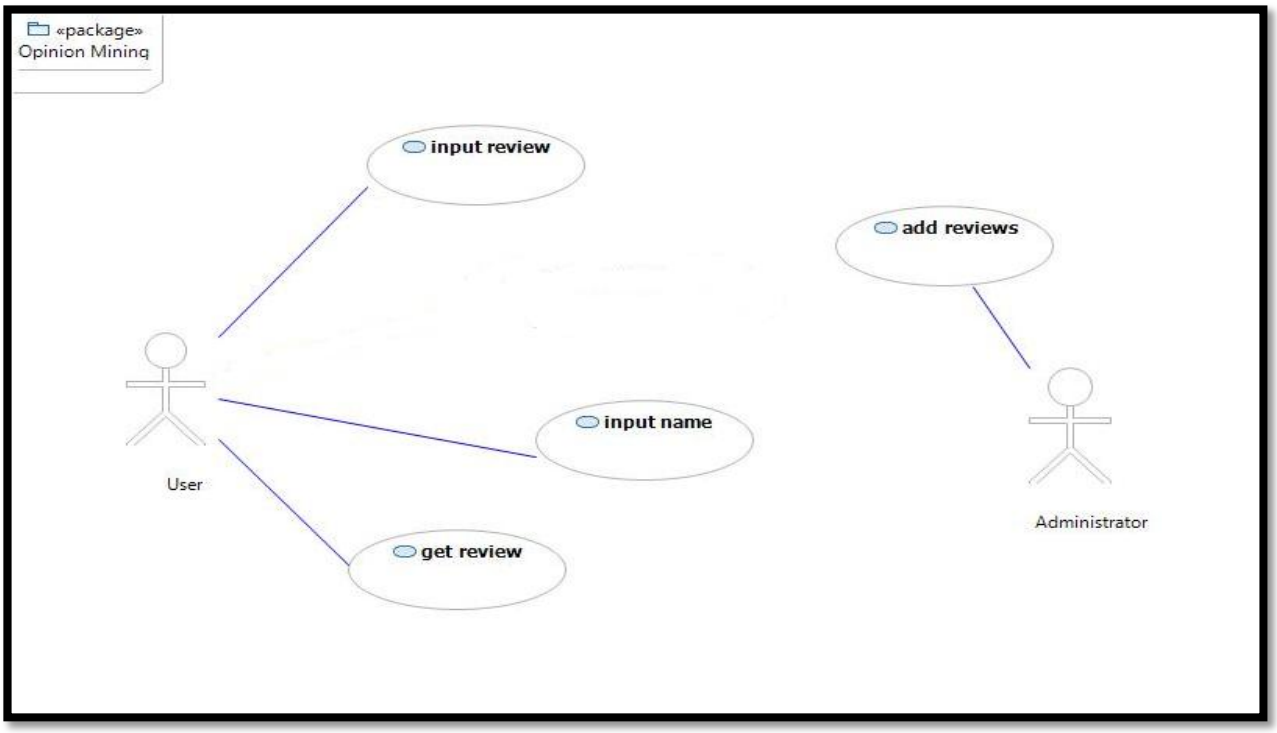
Advantages:

- Distribution of data is a straightforward
- Makes effective use of the networked systems. May require cheaper hardware.

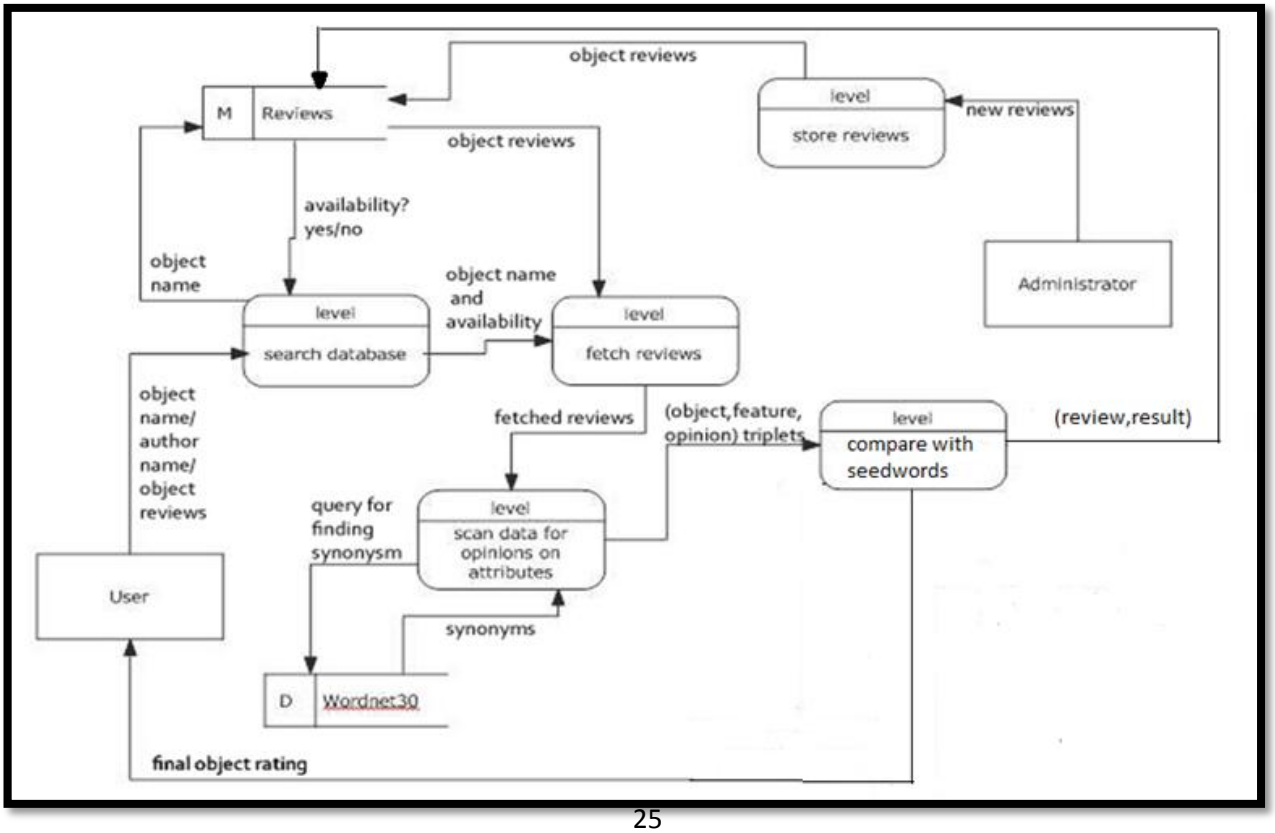
- Ease of the management- Upgrading, updates and maintenance of shared resources means that you only have to upgrade on the server instead of each and the every individual PC. Having to backup data from the server is easier than having to the backup multiple PCs
- Sharing of resources. All users can utilize the resources on server.

3.3. Software design

3.3.1. Use case diagram (Fig. 6):



3.3.2. Data Flow diagram (Fig. 7):



3.3.3. Data Base Schema:

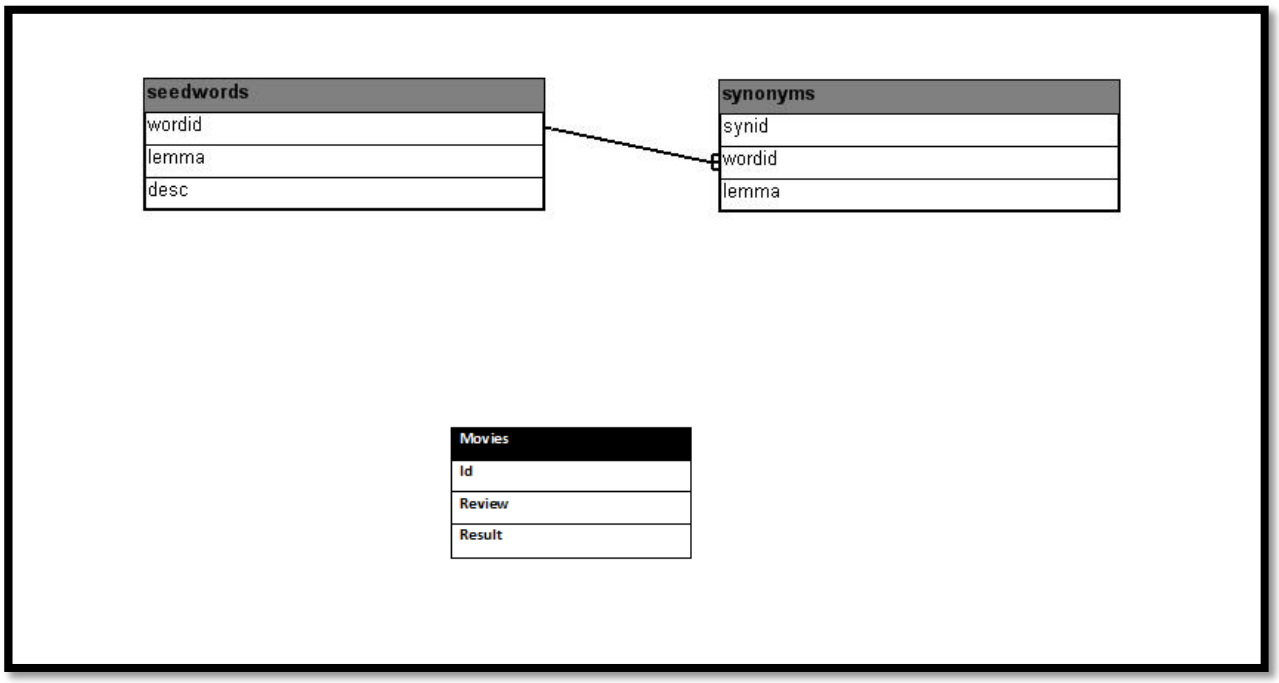


Fig. 8

3.4. System requirement

Hardware requirements:

- CPU : 1.6 GHz
- RAM : 512 MB
- Display : 1024*768 Monitor
- Hard Disk : 16 GB

Software requirements:

- OS : Windows
- Software : My SQL, Netbeans

Programming language:

- JAVA

4. Implementation and Schedule of Activities:

4.1. Implementation

The project can be divided into the following tasks:

- **Downloading reviews** [10]**:** All the movie reviews are taken from the site www.stuff.mit.edu. User reviews where authors provide quantitative or binary ratings are perfect for training and testing a classifier for sentiment or the orientation. Test Data being used by us consists of set of 30000 movie reviews. Training Data consists of set of 1000 positive and 1000 negative reviews from the above set of reviews. Working with movie reviews as evaluative documents. Reasons behind this are:
    - They generally tend to be of a considerable length.
    - Many reviews of single movie are easily available.
- **Setting up Wordnet:** WordNet is large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing distinct concept. Synsets are interlinked by means of conceptual-semantic and the lexical relations. WordNet superficially resembles a thesaurus, in that it groups words together based on the meanings.
- **Stemming:** Stemming is the process for reducing inflected or sometimes derived from words to their stem, base or root form—generally written word form. The stem need not be identical to the morphological root of word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself valid root.

  (Write, wrote, written)- >write

  Stemming is useful while doing any kind of the text-analysis. When working with the contents of the text, the different types of verbs, and the different endings for singular and plural, make it difficult to discern the importance of specific words within the text, when each word is treated as usually it is.

- **Seed words:** The following Table 1 below shows the opinion words of the movie elements.

Opinion	**Keywords**
Positive	love, wonderful, best, great, superb, still, beautiful
Negative	bad, worst, stupid, waste, boring, ?, !

The list was generated by examinations of opinion word frequency counts in 700 positive, 700 negative movie reviews and selecting the 7 most frequent words for each category.

- By Bo Pang, Lillian Lee and Shivakumar Vaithyanathan in "Thumbs up? Sentiment Classification using Machine Learning Techniques" [7]

Because opinion words can vary greatly from author to author, we added the synonymic words of all above keywords to expand the keyword list. The final list consists of 185 opinionated words. The implementation of the stemmer is explained in the next chapter.

- **Comparison:** The reviews were the compared with the seed words. The total number of the positive as well as negative words was found out. Review was termed positive if positive words exceeded negative words and vice-versa. In case of tie the review was termed as neutral.

This is the main implementation of the project.

4.2. Schedule of activities

Task Name	Duration
Project Planning	1 week
Understanding the problem	1 Week
Literature Survey	2 Week
Requirement analysis	2 Week
Identifying Software Requirement	3 Week
UML modelling	1 Week
Development	2 Week
Collecting Evaluative Documents	1 Week
Formatting Documents	1 Week
Implementing Wordnet Connectivity	2 Week
Test Common Attribute Synonym Access	1 week
Identity Features and Attributes	1 week
Predict final rating of Output	1 week
Check Prediction Accuracy	1 week
Make appropriate changes	2 Week
Create user Interface	3 Week
Project Completion	1 week

Table 2

5. WordNet [8]

WordNet is large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relation. The resulting network of meaningfully related words and concept can be navigated with the browser. WordNet is also freely available for download. WordNet's structure makes it a helpful tool for computational linguistics and natural language processing.

WordNet superficially resembles thesaurus, in that it groups words together based on their meanings. However, there are some significant distinctions. First, WordNet interlinks not just word forms strings of letters but specific senses of words. As a result, words that are found in close proximity to one another in network are semantically disambiguated. Second, WordNet labels the semantic relations among words, whereas the grouping of words in thesaurus does not follow any explicit pattern other than meaning similarity.

Concepts, Words, Relations:

- Lexicon: labelling of concepts => words
- Humans label salient concepts
- Concepts differ in systematic ways: contrasts and similarities
- Consistent differences = relations
- If a few relations suffice to interlink most labelled concepts, then labelling is systematic (lexicon is regular)

There are two kinds of semantic relations:-

- Lexical (word-word) relations
- Conceptual (concept-concept) relations

The most frequently encoded relation between synsets is the super-subordinate relation (also called hyperonymy, hyponymy or ISA relation). It links more general synsets like {furniture, piece_of_furniture} to increasingly specific ones like bed and bunk bed . WordNet states that category furniture includes bed, which in turn includes bunked; conversely, concepts like bed and bunkbed make up the category furniture. All noun hierarchies ultimately go up to the root node {entity}. Hyponymy relation is transitive: if an armchair is a kind of chair, and if a chair is a type of furniture, then an armchair is a kind of furniture. WordNet distinguishes between Types (common

nouns) and Instances specific persons, countries and geographic entities. Thus, armchair is a type of a chair, Barack Obama is an instance of a president. Instances are always leaf terminal nodes in their hierarchies.

Metonymy, the part-whole relation holds between synsets like chair and back, backrest, seat and leg. Parts are inherited from their super ordinates: if chair has legs, then an armchair has legs as well. Parts are not inherited "upward" as they may be characteristic only of specific kinds of the things rather than the class as a whole: chairs and kinds of chairs have legs, but not all kinds of furniture have legs.

Verb synsets are arranged into hierarchies as well; verbs towards the bottom of the tree troponyms express increasingly specific manners characterizing an event, as in {communicate}-{talk}-{whisper}. The specific manner expressed depends on the semantic field; volume as in the example above is just one dimension along which verbs can be elaborated. Others are speed (move-jog-run) or intensity of emotion like-love-idolize. Verbs describing events that necessarily and unidirectional entail one another are linked: buy-pay, succeed-try, show-see, etc.

Adjectives are organized in the terms of antonym. Pairs of direct antonyms like wet dry and young-old reflect the strong semantic contract of their members. Each of these polar adjectives in turn is linked to a number of semantically similar ones: dry is linked to parched, arid, desiccated and bone-dry and wet to soggy, waterlogged, etc. Semantically similar adjectives are indirect antonyms of the control member of the opposite pole. Relational adjectives "pertainyms" point to the nouns they are derived from (criminal-crime).

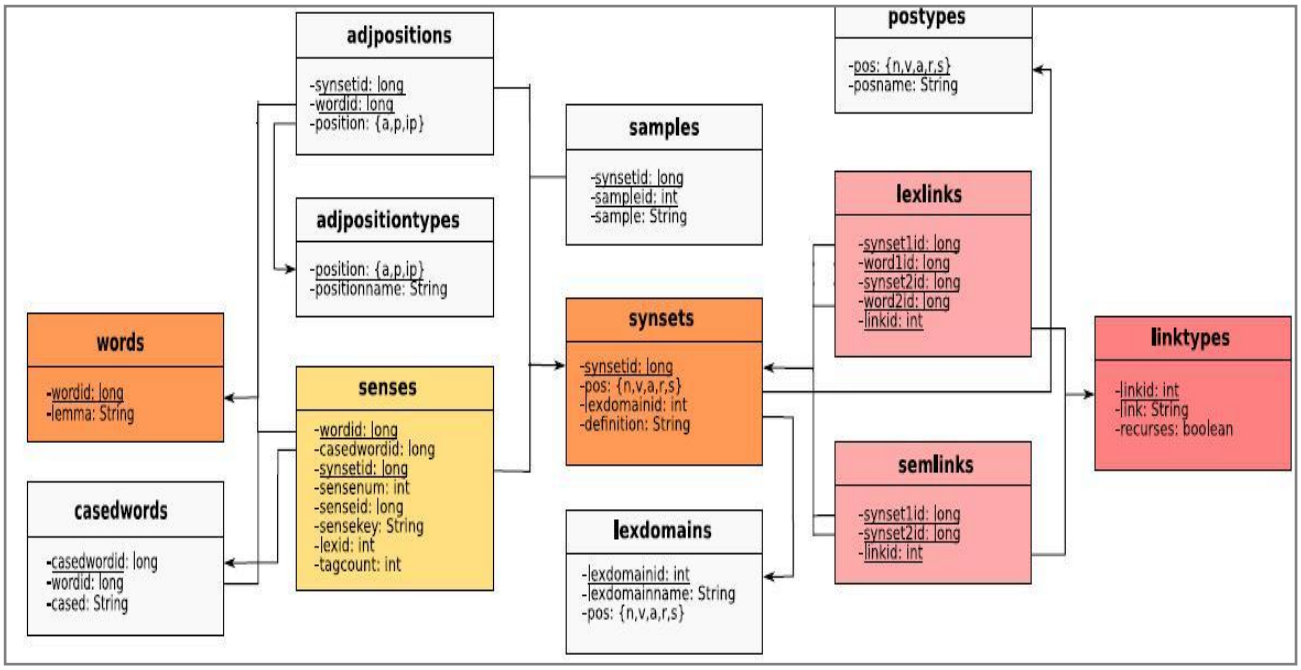
There are only few adverbs in WordNet e.g hardly, mostly, really, etc. as the majority of English adverbs are straightforwardly derived from adjectives via morphological affixation surprisingly, strangely, etc.

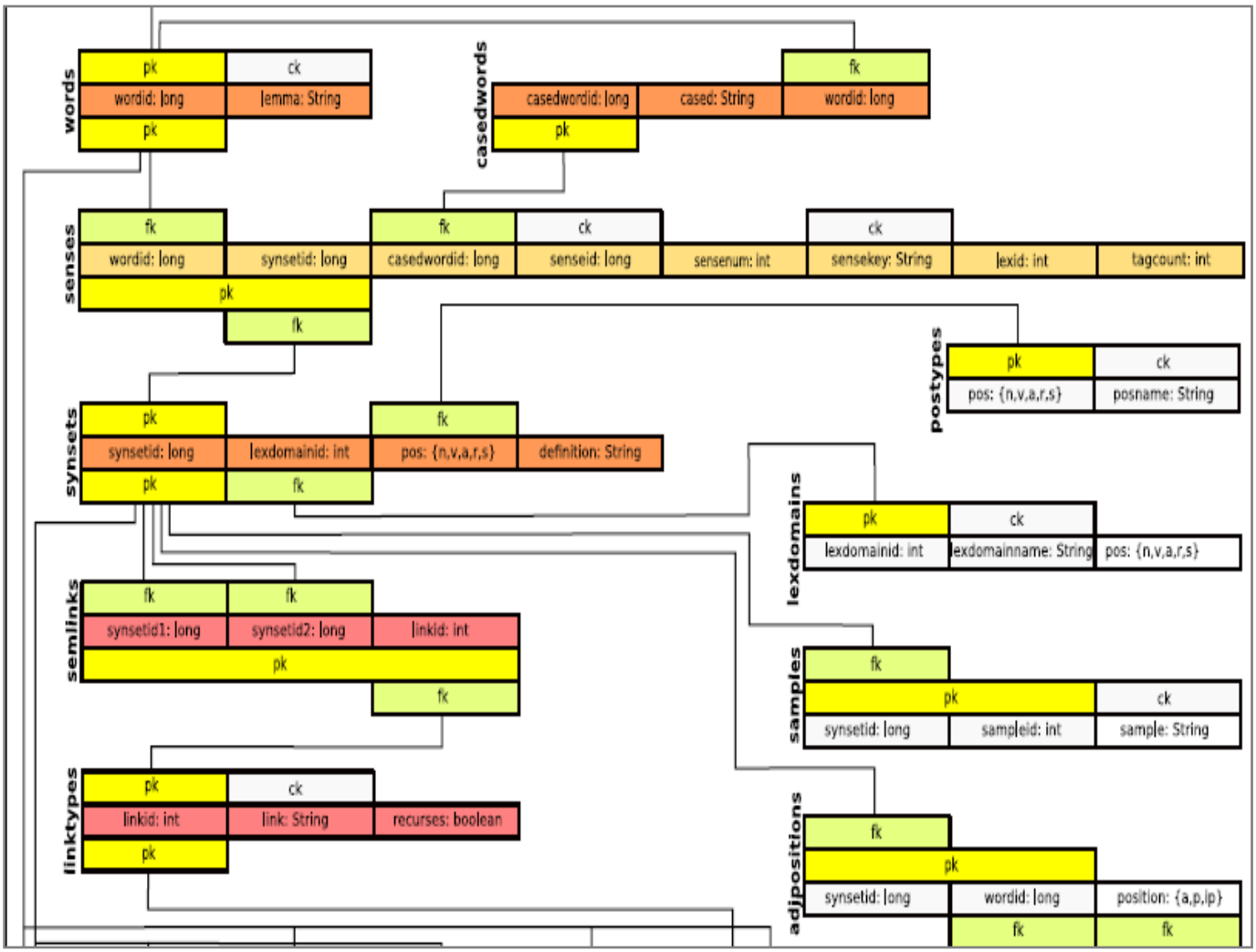
Cross-POS relations:

The majority of the WordNet's relations connect words from same part of speech (POS). Thus, WordNet really consists of the four sub-nets, one each for nouns, verbs, adjectives and adverbs, with few cross-POS pointers. Cross-POS relations include the "morph semantic" links that holds among semantically similar words sharing a stem with the same meaning: observe (verb), observant (adjective) observation, observatory (nouns). In many of the noun-verb pairs the semantic role of the noun with respect to the verb has been specified: sleeper, sleeping car is the LOCATION for {sleep} and {painter}is the AGENT of {paint}, while {painting, picture} is its RESULT.

Wordnet Design:

Fig. 9





Applications: WordNet has been used for a number of the different purposes in information systems, including word sense disambiguation, information retrieval, automatic text classification, automatic text summarization, and even automatic crossword puzzle generation.

Another prominent example of use of WordNet is to determine the similarity between words. Various algorithms have been proposed, and these includes considering the distance between the conceptual categories of words, as well as considering the hierarchical structure of the WordNet ontology.

The goal of WordNet was to develop a system that would be consistent with the knowledge acquired over the years about how human beings practice language. Anomic aphasia, for example, creates a condition that seems to selectively encumber individuals' ability to name objects; this makes the decision to partition the parts of speech into distinct hierarchies more of a principled decision than an arbitrary.

Problems and implementation: Unlike the other dictionaries, WordNet does not include information about etymology, pronunciation and the forms of irregular verbs and contains only limited information about usage. Though WordNet contains sufficiently wide range of common words, it does not cover special domain vocabulary. Since it is the primarily designed to act as an underlying database for different applications, those applications cannot be used in specific domains that are not covered by WordNet.

- **Implementation:** The WordNet library was dumped into database and used.

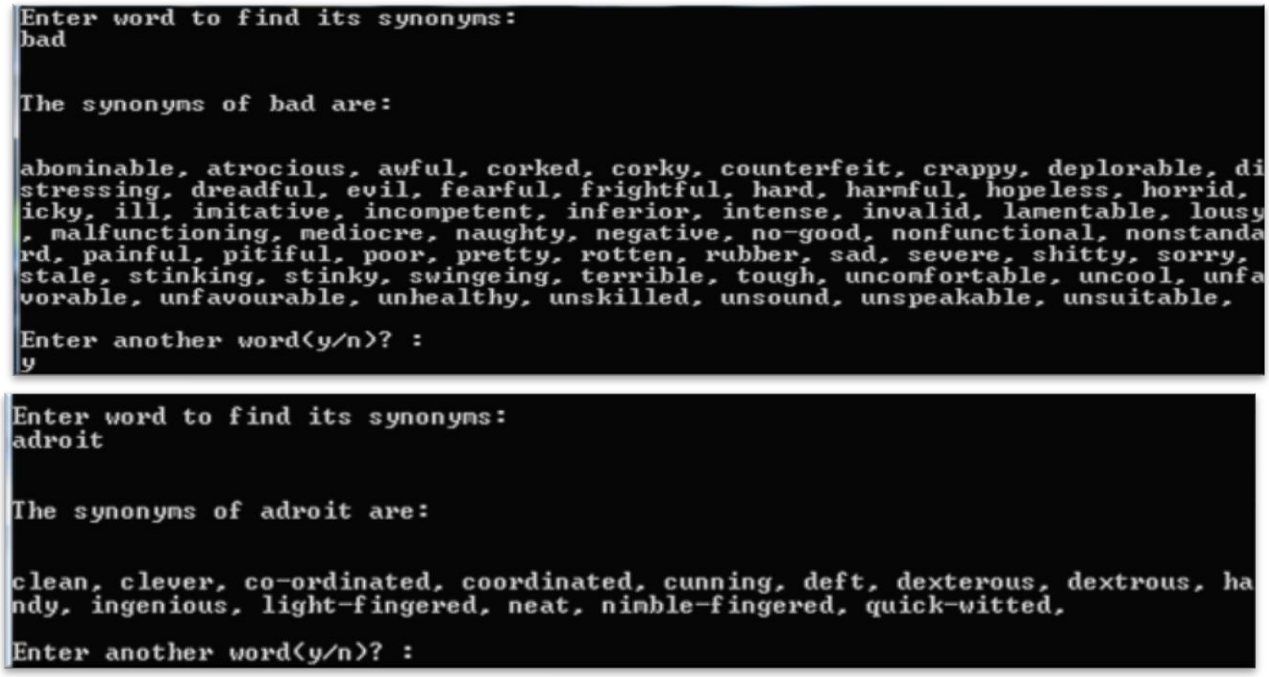


Fig. 10

- **Stemmer**[9]**:**

In linguistic morphology and information retrieval, **stemming** is the process for reducing inflected or sometimes derived words to their stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself valid root. Algorithms for stemming have been studied in the computer science since 1968. Many search engines treat words with the same stem as synonyms as a kind of query broadening, a process called the conflation.

Stemming programs are commonly referred to as stemming algorithms or the stemmers.

A stemmer for English, for example, should identify the string "cats" (and possibly "catlike", "catty" etc.) as based on the root "cat", and "stemmer", "stemming", "stemmed" as based on the "stem". A stemming algorithm reduces words "fishing", "fished", "fish", and "fisher" to the root word, "fish".

Brute Force Approach:

Unfortunately, stemming is a problem to do algorithmically, due to different rules and special cases in english language. An easier way is to stem brute by brute force, that is to use dictionary that lists all the words together with their stems.

One such dictionary, which is freely available, is the WordNet. And some nice people created an open source project that provides nice Java API to the dictionary, named JWNL.

JWNL API:

Using JWNL is too simple. First, call JWNL.initialize() somewhere in the initialization code of our program. Then, just call Dictionary.getInstance() to get currently installed dictionary. The only dictionary methods you should really ever need to call are the lookupIndexWord(), lookupAllIndexWords(), and getIndexWordIterator().

The other methods you may be interested in Relationship.findRelationships(), and those in the PointerUtils.

Relationship.findRelationships() allows you to find relationships of the given type between the two words (such as ancestry). Another way of thinking of a relationship is as the path from the source synset to the target synset.

The methods in PointerUtils allow you to find chains of pointers of given type. For example, calling PointerUtils.getHypernymTree() on the synset that contains "dog," returns a tree with all its parent synsets ("canine"), and its parents' parents ("carnivore"), etc., all the way to root synset ("entity").

JWNL provides support for accessing the WordNet database through three structures - the standard file distribution, a database, or in-memory map. Utilities are provided to convert from the file

structure to an SQL database or in-memory map, and a configuration file controls which system the library uses.

Output:

When a single word is passed:

```
C:\java>java StemmerTest writing
Dec 4, 2011 7:30:37 PM net.didion.jwnl.util.MessageLog doLog
INFO: Installing dictionary net.didion.jwnl.dictionary.FileBackedDictionary@131f
71a
write
```

When a review is passed:

Input Text	Output Text
Most good films will engage you with their characters and their drama, but how many can claim to challenge you at the same time? Oye Lucky Lucky Oye, directed by Dibakar Banerjee, is a film that keeps you on your toes; it's a film that never spoon-feeds you, instead expects you to read between the lines, to fill in the gaps for yourself, and to decode the subtext. It's true, Oye Lucky Lucky Oye is the kind of film that expects as much of its audience as its audience expects of the film.	most good film will engage you with their character and their drama but how many can claim to challenge you at the same time ? oye lucky lucky oye, direct by dibakar banerjee, be a film that keep you on your toe s a film that never spoonfeed you, instead expect you to read between the line to fill in the gap for yourself, and to decode the subtext . s true oye lucky lucky oye be the kind of film that expect a much of its audience a its audience expect of the film .

Fig. 11

6. Result and Conclusion

6.1. Result

The algorithm when run on a single review gave the following result:

Fig. 12

The algorithm, when implemented on pre-classified data, gave the following results:

- Results with 1000 positive movie reviews:
    - Number of correct predictions=685
    - Number of incorrect predictions=315
    - **Accuracy= 68.5%**
- Results with 1000 negative movie reviews:
    - Number of correct predictions=567
    - Number of incorrect predictions=433
    - **Accuracy= 56.7 %**
- **Overall Accuracy=62.6%**

It must be noted that the tie rates-percentage of documents where these two sentiments were rated equally likely-are quite high.

Ties were considered as **INCORRECT** prediction.

6.2. Conclusion

By analyzing our results we find that positive reviews have a better accuracy to be correctly found out than the negative reviews. This implies that the list of positive seed words are better than the negative seed words. We have achieved an overall accuracy of 62.6 % accuracy but this only a little more than flipping a coin and deciding the correct sentiments. This is because movies tend to deviate into the plots. The sum of the parts may not be sum of whole. For eg. a dark or a scary plot will tend to get a negative score even though movie may be good. Similarly terms may mean different in different situations and therefore may lead to a wrong result. Scary movie may be good for a horror movie but may be negative for love story.

6.3. Future Work

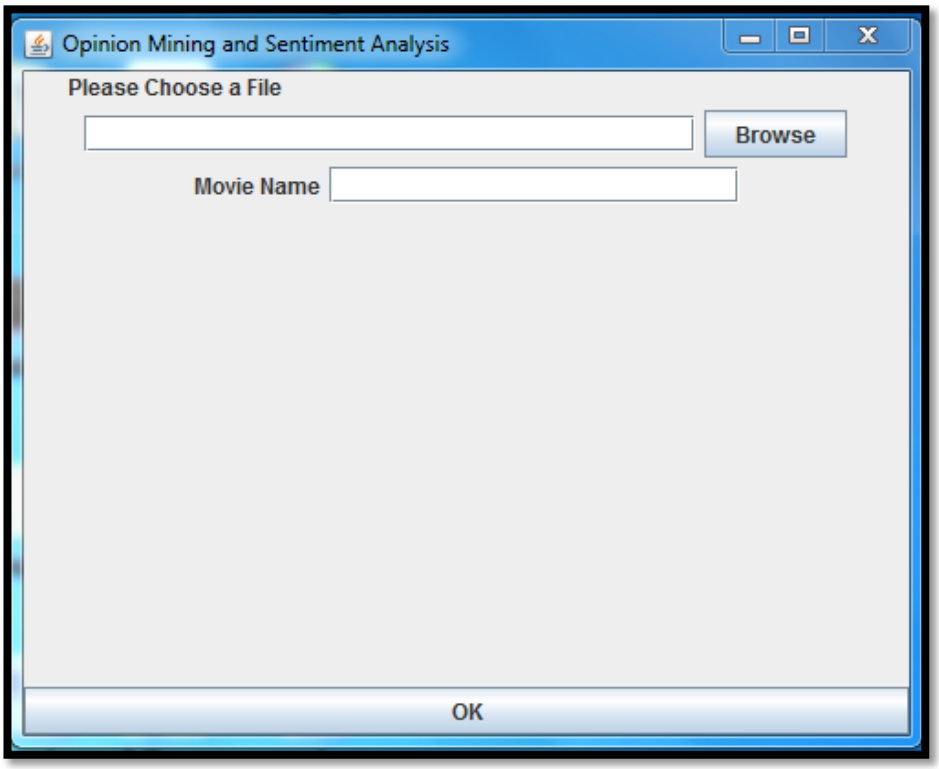
- Implementing a Machine Learning algorithm for dynamic update of opinion word list while scanning the future reviews.
- Using dependency relation templates to detect path between each feature word and each opinion word.

  Eg: If there is a negation relation, the opinion class can be transferred according to the simple rules:

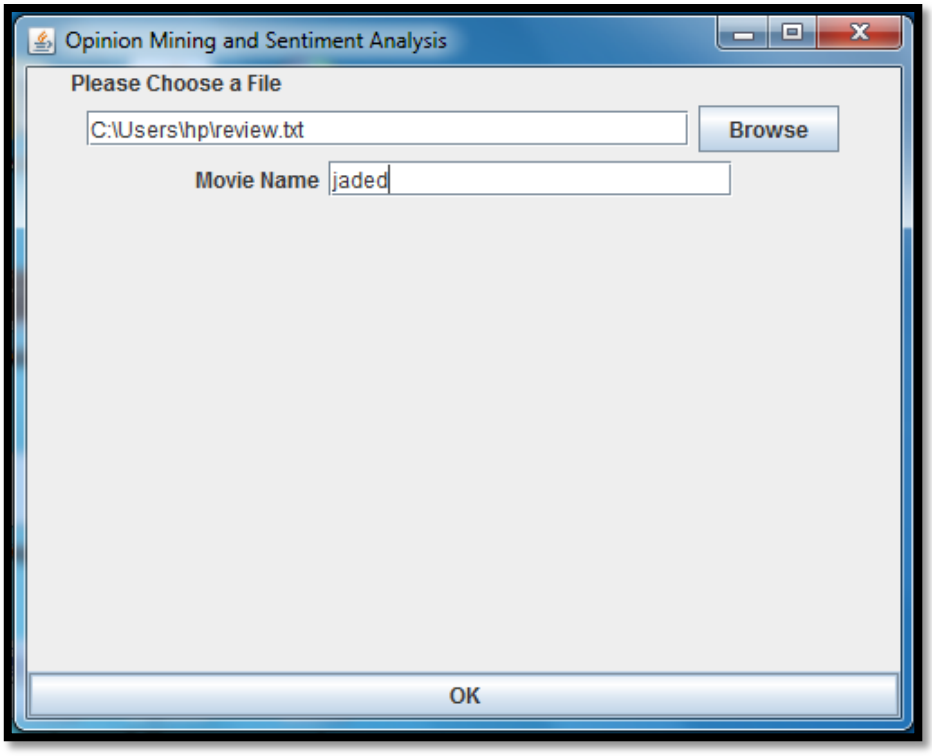
7. Snapshots and Code

7.1. Snapshots

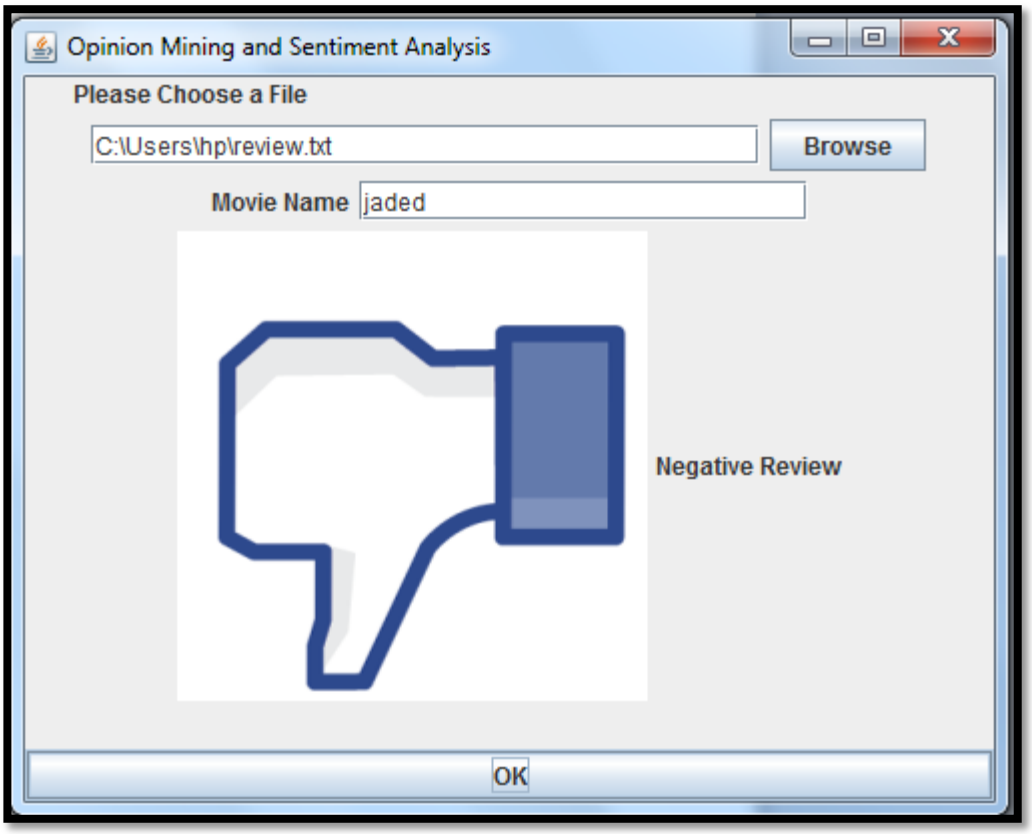
- Fig. 13: This is the software display. It has only two text boxes. One for the path of the review text file and the other for the movie name.



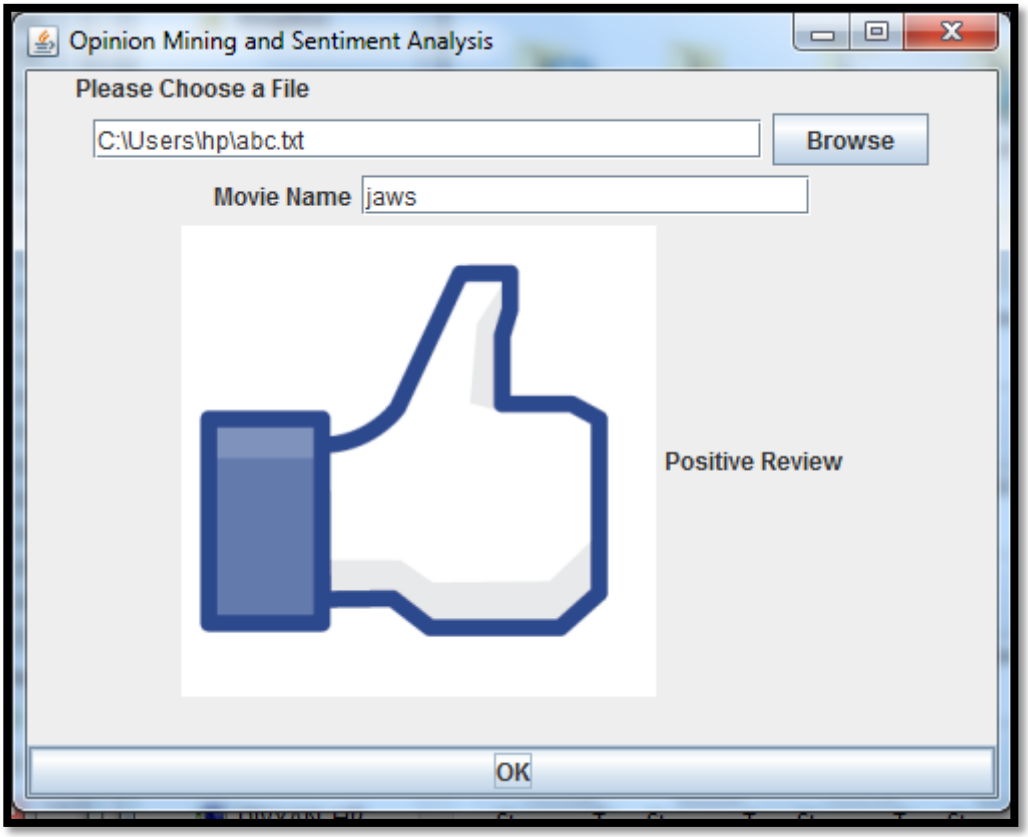
- Fig. 14: The user in this entered in this case the movie jaded which has an imdb rating 5.3 which is obviously poor.



- Fig. 15: The software returns a dislike for the movie affirming that it is a bad movie.



- Fig. 16: Similarly for Jaws it returns a positive feedback.



7.2. Code

Stemmer.java:

```
package javaapplication1;


import net.didion.jwnl.*;
import net.didion.jwnl.data.*;
import net.didion.jwnl.dictionary.*;


import java.io.FileInputStream;
import java.io.FileNotFoundException;


class Stemmer
{
        private int MaxWordLength = 50;
        private Dictionary dic;
        private MorphologicalProcessor morph;
        private boolean IsInitialized = false;
        public Stemmer ()
        {
                try
                {
                        JWNL.initialize(new FileInputStream("JWNLproperties.xml"));
```

```
                dic = Dictionary.getInstance();
                morph = dic.getMorphologicalProcessor();
                IsInitialized = true;
            }
            catch ( FileNotFoundException e )
            {
                System.out.println ( "Error initializing Stemmer: JWNLproperties.xml not found" );
            }
            catch ( JWNLException e )
            {
                System.out.println ( "Error initializing Stemmer: "
                        + e.toString() );
            }
          catch(NoClassDefFoundError e){
          }
        }
        public void Unload ()
        {
            dic.close();
            Dictionary.uninstall();
            JWNL.shutdown();
        }
        public String StemWordWithWordNet ( String word )
```

```
{
        if ( !IsInitialized )
                return word;
        if ( word == null ) return null;
        if ( morph == null ) morph = dic.getMorphologicalProcessor();

        IndexWord w;
        try
        {
                w = morph.lookupBaseForm( POS.VERB, word );
                if ( w != null )
                        return w.getLemma().toString ();
                w = morph.lookupBaseForm( POS.NOUN, word );
                if ( w != null )
                        return w.getLemma().toString();
                w = morph.lookupBaseForm( POS.ADJECTIVE, word );
                if ( w != null )
                        return w.getLemma().toString();
                w = morph.lookupBaseForm( POS.ADVERB, word );
                if ( w != null )
                        return w.getLemma().toString();
        }
        catch ( JWNLException e )
        {
```

```
                }
                return null;
        }
        public String Stem( String word )
        {
                String stemmedword;
                stemmedword = StemWordWithWordNet (word);
                if ( stemmedword != null )
                {
                        return stemmedword;
                }
                return word;
        }
}
```

Op_mining.java:

```
package javaapplication1;



import java.io.*;
import java.sql.*;


class Op_mining
{
        public int omsa(String filename,String moviename)throws Exception
```

```
        {
        //Sql part below
        String words[][]= new String[300][3];
      String temp, name=null;
        int x=0,i,seedwordcount,totwordcount,result;
        DataInputStream d=new DataInputStream(System.in);
        //try
        //{
                Class.forName("com.mysql.jdbc.Driver");
      Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/wordnet30","root","wnsql");
                Statement stmt=con.createStatement();
                ResultSet rs = stmt.executeQuery("select * from wordnet30.seedwords;");
                while(rs.next())
                {
                        words[x][0]=rs.getString(1);
                        words[x][1]=rs.getString(2);
                        words[x][2]=rs.getString(3);
//                      System.out.print(words[x][1]+","+words[x][2]+"\n");
                        x++;
                }
                seedwordcount=x;
                rs=stmt.executeQuery("select * from wordnet30.synonyms;");
                while(rs.next())
```

```
                {
                        temp=rs.getString(2);
                        for(i=0;i<seedwordcount;i++)
                        {
                                if(temp.compareTo(words[i][0])==0)
                                {
                                        words[x][2]=words[i][2];
                                        break;
                                }
                        }
                        words[x][1]=rs.getString(3);
                        words[x][0]=rs.getString(1);
//                      System.out.print(words[x][1]+","+words[x][2]+"\n");
                        x++;
                }
        //}catch(Exception e){}
        totwordcount=x;
// Stemmer part below
                String ftext=". ";
                Stemmer stem=new Stemmer();
                int c=0;
                i=0;
                char buffer[]=new char[40];
                FileInputStream fin;
```

```
BufferedWriter out;
try
{
        try
        {
                fin=new FileInputStream(filename);
        }catch(FileNotFoundException e)
        {
                System.out.println("File not found");
                return 100;
        }
        try
        {
                out = new BufferedWriter(new FileWriter("s.txt"));
        }catch(FileNotFoundException e)
        {
                System.out.println("Error opening file");
                return 100;
        }
}catch(ArrayIndexOutOfBoundsException e)
{
        System.out.println("usage: Showfile");
        return 100;
}
```

```
while(i!=-1)
{
        i=fin.read();
        if(i>=65 && i<=90)
                i+=32;
        if((char)i=='\n' || (char)i=='\r')
                i=32;
        if(i!=32 && (char)(i)!='.' && (char)(i)!='?' &&        (char)(i)!='!')
        {
                        buffer[c++]=(char)i;
        }
        else
        {
        //      System.out.println(stem.Stem(new String(buffer,0,c)));
                out.write(stem.Stem(new String(buffer,0,c)));
                ftext=ftext+stem.Stem(new String(buffer,0,c));
                c=0;
        //      System.out.println(Character.toString((char)(i)));
                if(i!=32)
                {
                        out.write(Character.toString(' '));
                        ftext=ftext+Character.toString(' ');
                }
                out.write(Character.toString((char)(i)));
```

```
                        ftext=ftext+Character.toString((char)(i));
                    }
                }
        //      System.out.println(stem.Stem(new String(buffer,0,c-1)));
                out.write(stem.Stem(new String(buffer,0,c-1)));
                ftext=ftext+stem.Stem(new String(buffer,0,c-1));
                fin.close();
                out.close();
//              System.out.println(ftext);
//Comparison from here

                int li=ftext.lastIndexOf(". "),wc,wi,wli,pc=0,nc=0;
                String sent,word;
                c=1;
                while(c<=li)
                {
                    i=ftext.indexOf(". ",c);
                    sent=ftext.substring(c,i);
                    wli=sent.lastIndexOf(' ');
                    wc=1;
//                  System.out.println(sent+"\n\n");

                    while(wc<=wli)
                    {
```

```
                    wi=sent.indexOf(' ',wc);
                    word=sent.substring(wc,wi);

                    for(x=0;x<totwordcount;x++)
                    {
                        if(word.compareTo(words[x][1])==0)
                        {
                            System.out.println(word+": "+words[x][2]);
                            if(words[x][2].compareTo("p")==0)
                                pc++;
                            else if(words[x][2].compareTo("n")==0)
                                nc++;
                        break;
                        }
                    }
                    wc=wi+1;
                }
                c=i+1;
            }
   System.out.println("\nTotal +ve words="+pc);
   System.out.println("Total -ve words="+nc);
   if(pc>nc){
       result=1;}
else if(nc>pc){
```

```
            result=-1;}
        else{
            result=0;}
        //adding movie in database
        PreparedStatement pstmt = con.prepareStatement("INSERT INTO wordnet30.movies (name, review,result) VALUES (?, ?,?);");
            pstmt.setString(1, moviename);
            pstmt.setString(2, ftext);
            pstmt.setInt(3,result);
            pstmt.executeUpdate();
            return result;
          }
}
```

Gui.java:

```
package javaapplication1;


import java.awt.BorderLayout;
import java.awt.Component;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import javax.swing.*;
public class Gui implements ActionListener{
```

```
JFrame mainf;
JPanel p1;
JButton browse,ok;
JTextField address,movie;
JFileChooser chooser;
public void mainFrame(){
    mainf=new JFrame("Opinion Mining and Sentiment Analysis");
    mainf.setBounds(400, 200, 500, 400);
    mainf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    mainf.setLayout(new BorderLayout());

    p1=new JPanel();

    JLabel l=new JLabel("        Please Choose a File");
    address=new JTextField(30);
    address.setEditable(true);

    browse= new JButton("Browse");
    ok=new JButton("OK");
    JLabel l1=new JLabel("Movie Name");
    movie=new JTextField(20);
    p1.add(address);
    p1.add(browse);
    p1.add(l1);
```

```
    p1.add(movie);
    mainf.add(l,BorderLayout.NORTH);
    mainf.add(p1,BorderLayout.CENTER);
    mainf.add(ok,BorderLayout.SOUTH);
    browse.addActionListener(this);
    ok.addActionListener(this);

    mainf.setVisible(true);
}
public void actionPerformed(ActionEvent e) {
    if(e.getActionCommand().equals("Browse")){
        chooser=new JFileChooser();
        chooser.showOpenDialog(mainf);
        address.setText(chooser.getSelectedFile().getAbsolutePath());
    }
    if(e.getActionCommand().equals("OK")){
        if(address.getText().length()==0){
            JOptionPane.showMessageDialog(mainf, "No File Selected");
            return;
        }
        if(movie.getText().length()==0){
            JOptionPane.showMessageDialog(mainf, "No Movie Specified");
            return;
        }
```

```
String filePath=address.getText();
String moviename=movie.getText();
try{
   int i=new Op_mining().omsa(filePath,moviename);
   ImageIcon icon;
   JLabel msg;
   if(i==1){
      //JOptionPane.showMessageDialog(mainf, "Positive Review","Review",
       //      JOptionPane.INFORMATION_MESSAGE,new ImageIcon("like.png"));}
      icon=new ImageIcon("like.png");
      msg=new JLabel("Positive Review");
      msg.setIcon(icon);
   }
   else if(i==-1){
      icon=new ImageIcon("dislike.png");
      msg=new JLabel("Negative Review");
      msg.setIcon(icon);
   }
   else{
      icon=new ImageIcon("neutral");
      msg=new JLabel("Neutral Review");
      msg.setIcon(icon);
   }
   p1.add(msg);
```

```
            p1.validate();
            mainf.validate();
            mainf.repaint();
        }catch(Exception ex){}
    }
  }

  public static void main(String[] args){
    Gui m=new Gui();
    m.mainFrame();
  }
}
```

8. References

[1] Bing Liu, Opinion Mining. Department of Computer Science, University of Illinois at Chicago, 851 S. Morgan Street Chicago, IL 60607-0753, liub@cs.uic.edu

[2] Peter D. Turney, Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Institute for Information Technology, National Research Council of Canada, Ottawa, Ontario, Canada, K1A 0R6, peter.turney@nrc.ca

[3] Kushal Dave, Steve Lawrence and David M. Pennock. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. At NEC Laboratories America, Princeton, New Jersey.

[4] Scott Nowson, Scary Films Good, Scary Flights Bad,
Topic Driven Feature Selection For Classification Of Sentiment. Appen Pty. Ltd, 1 Railway Street, Chatswood, Sydney, Australia, snowson@appen.com.au

[5] Li Zhuang, Feng Jing and XiaoYan Zhu. Movie Review Mining and Summarization. At Microsoft Research Asia and State Key Laboratory of Intelligent Technology and Systems. In *CIKM'06,* November 5–11, 2006, Arlington, Virginia, USA.

[6] Bo Pang(*Yahoo! Research)* and Lillian Lee(*Computer Science Department, Cornell University*) Opinion mining and sentiment analysis.

[7] Bo Pang and Lillian Lee(Department of Computer Science, Cornell University) and Shivakumar Vaithyanathan(IBM Almaden Research Center). Thumbs up? Sentiment Classification using Machine Learning Techniques

[8] http://wordnet.princeton.edu/ -The Wordnet website.

[9] - http://www.sourceforge.net/projects/jwordnet/ - Java API for accessing the WordNet relational dictionary by the JWNL development team

[10] www.stuff.mit.edu – Compiled movie reviews