**Chapter 4**

**SUMMARIZING THE CUSTOMERS REVIEW**

**4.1 DFD Diagram**

Crawl Reviews

Review

Database

Frequent Features Generation (TF-IDF)

POS-Tagging

Indexing

Adjective Extraction

Index

Database

Opinion word Extraction

Opinion word Extraction

TF-IDF POS-Tagging

Summary

The system performs the summarization in three main steps (as discussed before): As indicated above, our task is performed in three main steps:

(1) Mining product features that have been commented on by customers. We make use of both data mining and natural language processing techniques to perform this task.

(2) Identifying opinion sentences in each review and note that these opinion sentences must contain one or more product features identified above.

(3)A set of adjective words (which are normally used to express opinions) are identified using a natural language processing method. These words are also called *opinion words*. Then the opinion sentences are identified that contain those adjective words.

(4) Intersecting both step (2) and (3) for Summarizing the results.

Given the inputs, the system first downloads (or crawls) all the reviews, and put them in the review database. It then finds those “hot” (or frequent) features that many people have expressed their opinions on using the TF-IDF technique. After that, the opinion words are extracted using the resulting frequent features. On the other hand Part of Speech tagging is applied to find those sentences that contain adjective words. Then we intersect both set of opinion sentences to generate the summary.

Steps to summarize the reviews are as follows:

**4.2 Crawl Reviews and Review Database:**

A crawler is a program that visits Web sites and reads their pages and other information in order to create entries for a [search engine](http://searchsoa.techtarget.com/definition/search-engine) index. The major search engines on the Web all have such a program, which is also known as a "spider" or a "bot." Crawlers are typically programmed to visit sites that have been submitted by their owners as new or updated. Entire sites or specific pages can be selectively visited and indexed. Crawlers apparently gained the name because they crawl through a site a page at a time, following the links to other pages on the site until all pages have been read.

In Our project the task of crawler is to visit those pages of flipcart.com that contains reviews. It then crawl those pages and extract reviews and product name from that page. This data is then stored into the file. Each product have their individual file that contains all reviews commented by the customer.

The review database and file containing those reviews looks as shown below:

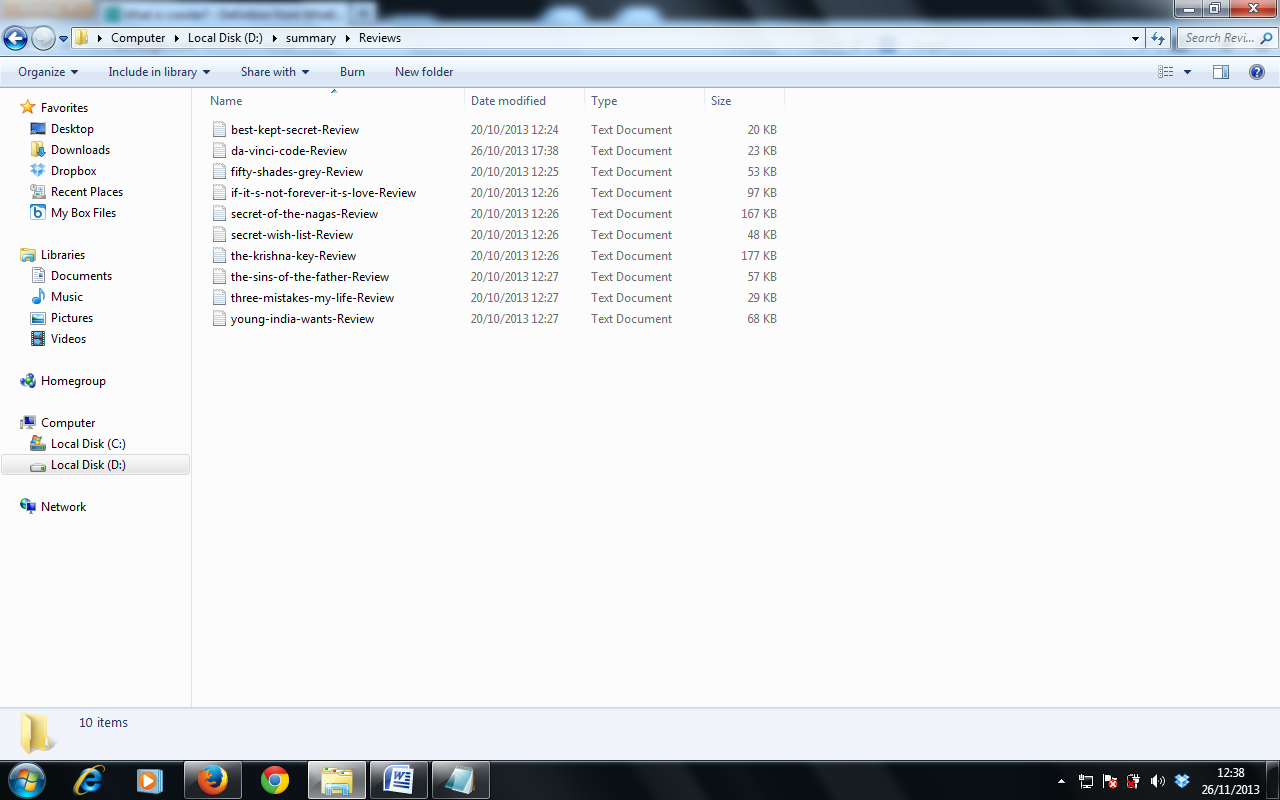


Fig: review database

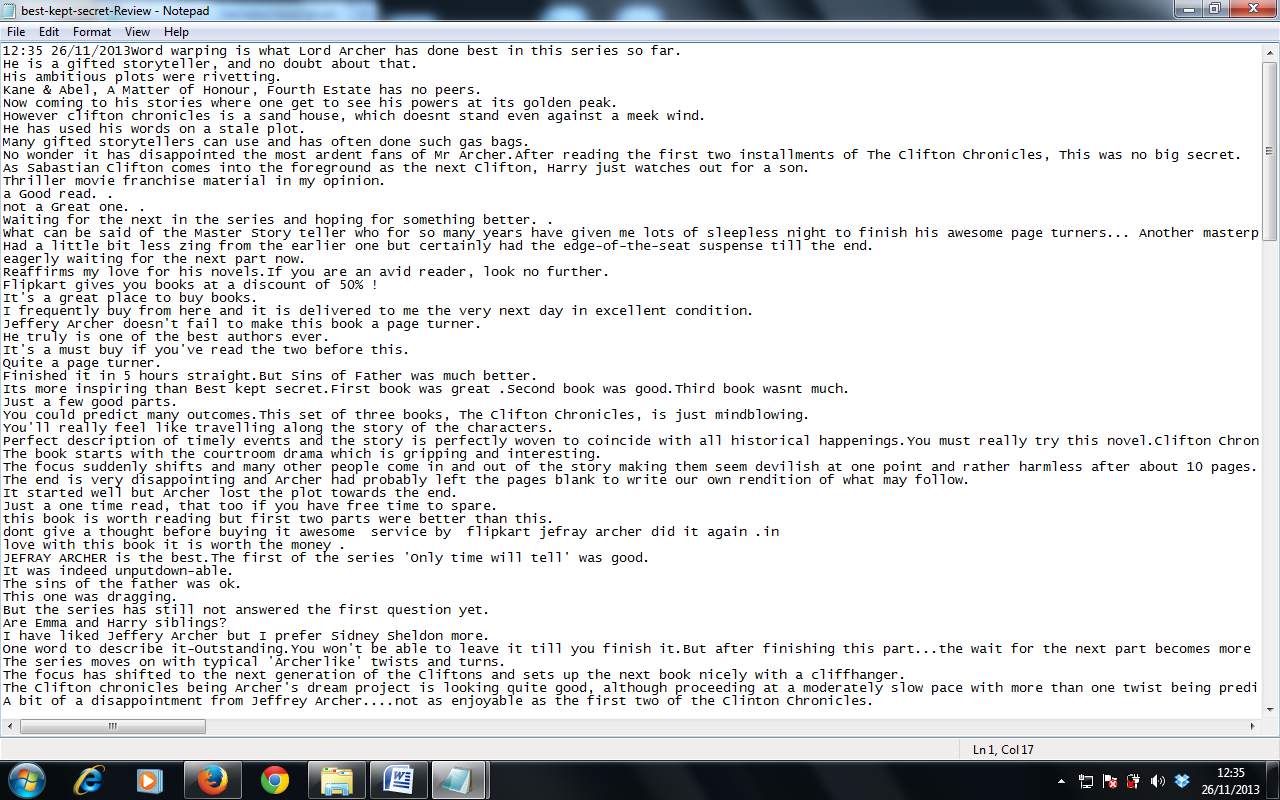


Fig: reviews of best kept secret book

**4.3 Indexing**

The Snap shot of complete Index is as follow:

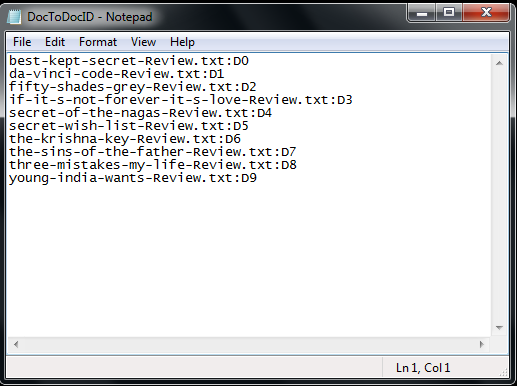


Fig: Index of Document to document ID

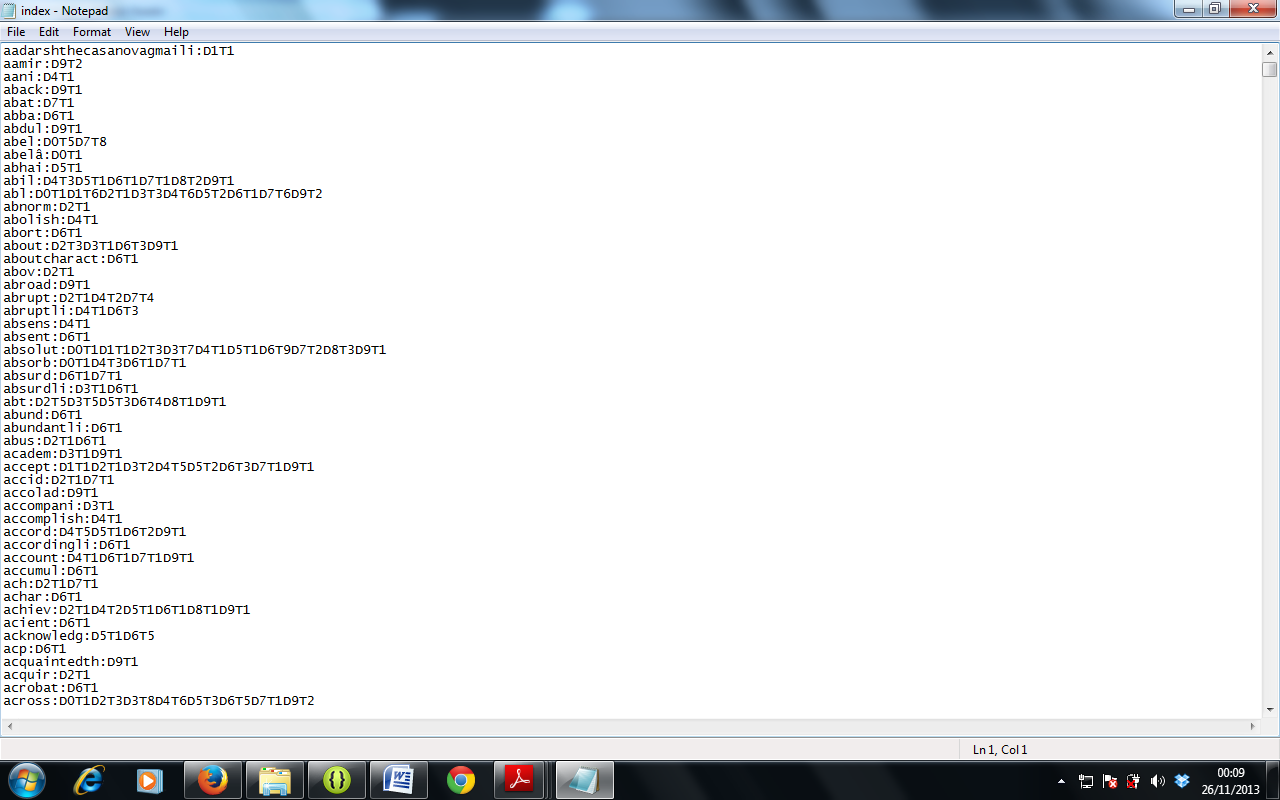


Fig: Index

**4.4 Frequent Features Generation**

With such a feature-based summary, a potential customer can easily see how the existing customers feel about the digital camera. If he/she is very interested in a particular feature, he/she can drill down by following the <individual review sentences> link to see why existing customers like it and/or what they complain about. For a manufacturer, it is possible to combine

summaries from multiple merchant sites to produce a single report for each of its products.

Term (or feature) selection is necessary to reduce noise, as well as to prevent Over fitting. Some machine learning techniques exhibit inferior performance when presented with too many attributes , so it is essential to select only the best ones. Two main approaches for feature selection are filtering and wrapper model. The filtering approach receives a set of features, and filters it independently from the induction algorithm. The wrapper model searches for good feature subsets, and evaluates them using n-fold cross validation on the training data. This scheme may be used in conjunction with any induction algorithm, which is used for evaluating feature subsets on the validation set. The search for feature subsets can be performed using simple greedy algorithms such as backward elimination or forward selection, or more complex ones that can both add and delete features at each step.

Here, we focus on finding frequent features, i.e., those features that are talked about by many customers (finding infrequent features will be discussed later). For this purpose, we use TF-IDF to find all frequent item sets. In our context, an item set is simply a set of words or a phrase that occurs together in some sentences. The main reason for using TF-IDF is because of the following observation. It is common that a customer review contains many things that are not directly related to product features. Different customers usually have different stories. However, when they comment on product features, the words that they use converge. Thus using TF-IDF to find frequent item sets is appropriate because those frequent item sets are likely to be product features. Those noun/noun phrases that are infrequent are likely to be non-product features.

This sub-step identifies product features on which many people have expressed their opinion. Before discussing frequent feature identification, we first give some example sentences from some reviews to describe what kinds of opinions that we will be handling. Let us see some

Sentence from the reviews of a digital camera:

*“The pictures are very clear.”*

In this the customers talk about the picture quality of the camera. In our project we extract top ten frequent features using the formula of Term-frequency and Inverse Document Frequency.

Term-Frequency is considered as the total no. of times that term occur into the document and Inverse Document frequency is calculated for those features (terms) that are rare but considered as equally important. IDF is calculated using the formula:

IDF = 1+log (N/DF)

Where, N is the total no. of documents in a collection and DF is the Document Frequency.

The combination of TF-IDF helps to reduce the scope of most frequent terms like the name of product that are less important and increases the scope of those terms that are not so common but important. This helps us to find the top ten most important terms that are the important features of the product that the users talk about.

snapshort of one such product is as follows:

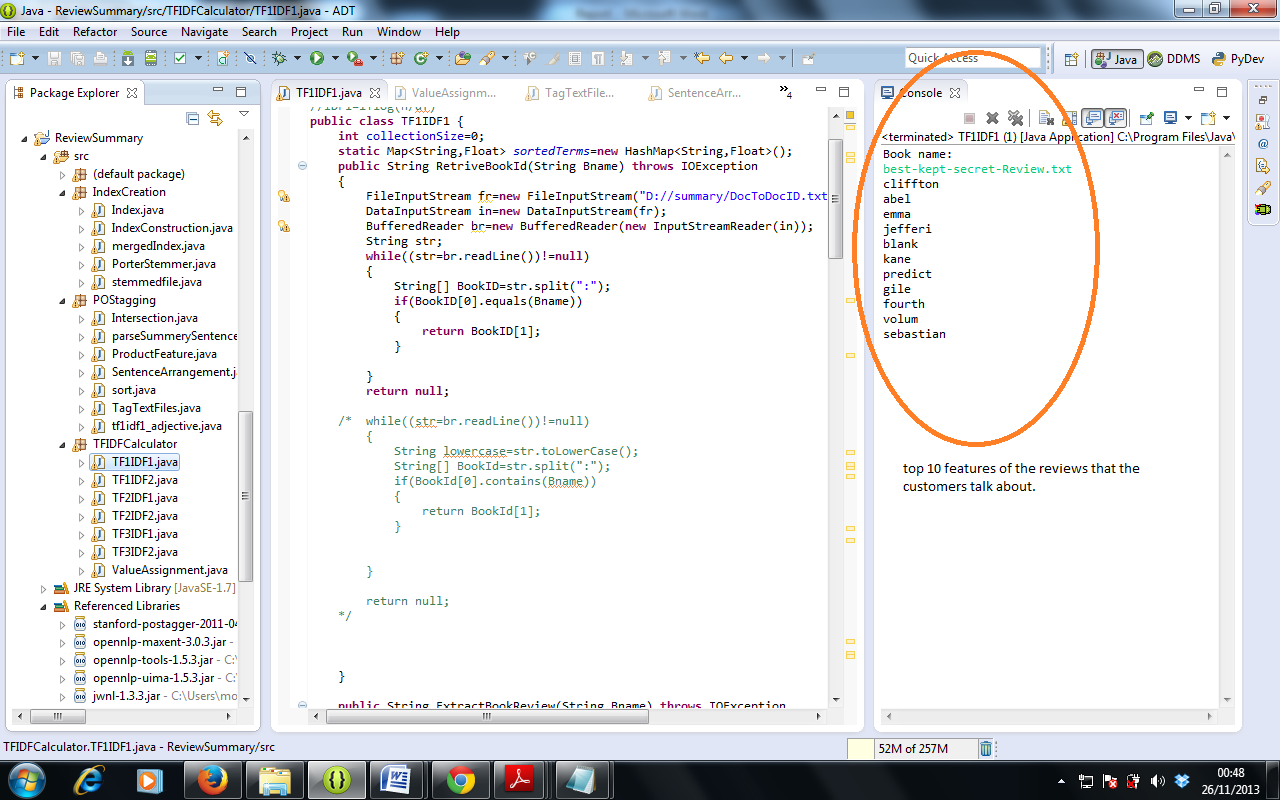


Fig: Top 10 features generated from reviews of a book named best kept secret.

**4.5 Opinion Word Extraction**

We now identify opinion words. These are words that are primarily used to express subjective opinions. Clearly, this is related to existing work on distinguishing sentences used to express subjective opinions from sentences used to objectively describe some factual information.

Let us first define an opinion sentence.

**Definition:** *opinion sentence*

If a sentence contains one or more product features and one or more opinion words, then the sentence is called an *opinion sentence.*

We extract opinion words in the following manner :

for each sentence in the review database

if (it contains a frequent feature)

Record that sentence as an opinion sentence.

For example, *strap* is a frequent feature and the sentence “*The strap is horrible and gets in the way of parts of the camera you need access to*.” In the review database is recorded as a opinion sentence. In this way a set is created that contains all the opinion sentences. This set is saved to use further for generating a summary.

Snap short the opinion sentences generated from the reviews of Best Kept Secret book are as follow:

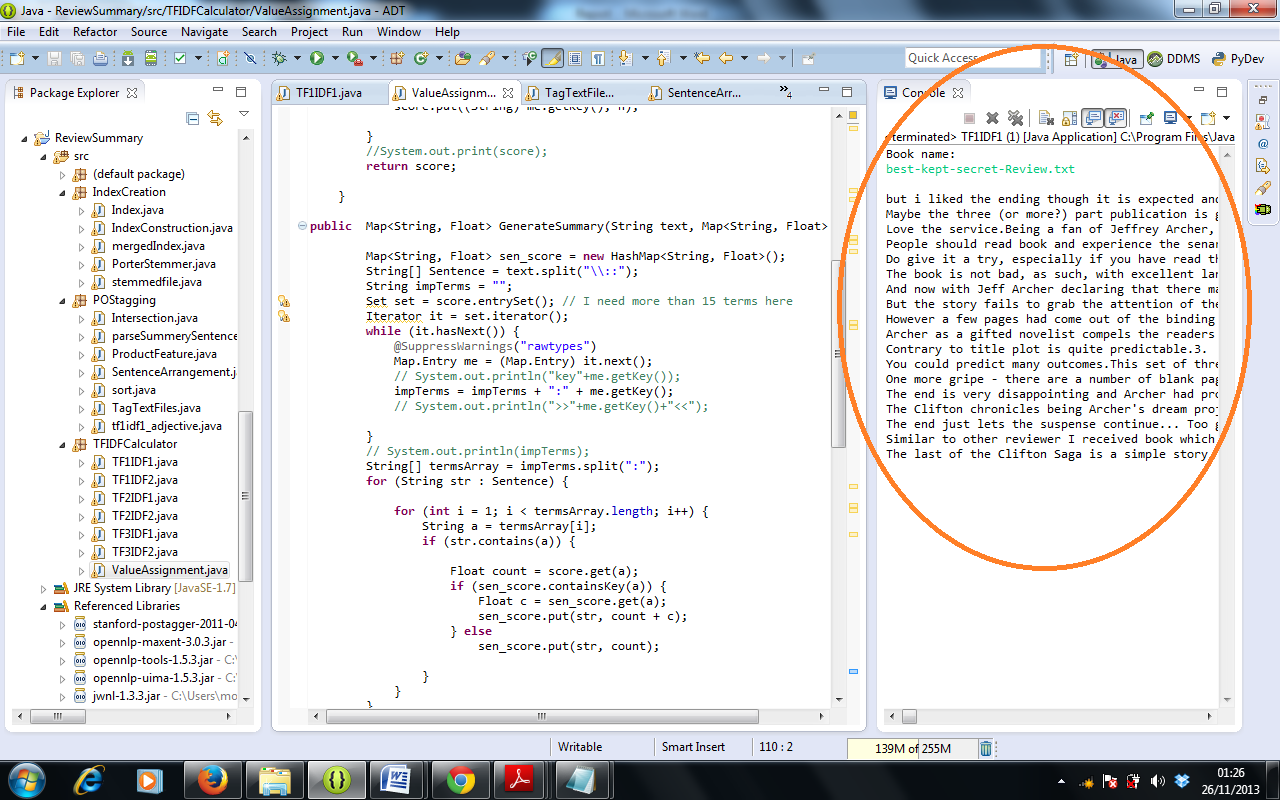


Fig: opinion sentences of a best kept secret book review.

**4.6 POS-Tagging**

Our systems aim is to find adjectives of those features that the users talk about. Thus the part-of-speech tagging is crucial. We use the Stanford-postagger parser, which parses each sentence and yields the part-of-speech tag of each word (whether the word is a noun, verb, adjective, etc) and identifies simple noun and verb groups . Before discussing the application of part-of-speech tagging from natural language processing, we first give some example sentences from some reviews to describe what kinds of opinions that we will handle. The following shows a sentence with the POS tags. Example: The only issue kept secret by the crook becomes known to readers and other characters in the book.

The/DT only/JJ issue/NN kept/VBD secret/JJ by/IN the/DT crook/NN becomes/VBZ known/VBN to/TO readers/NNS and/CC other/JJ characters/NNS in/IN the/DT book/NN

All tags have different meaning that is described below:

Adjective: JJ

Nouns: NN, NNP, NNS

Verbs: VBD, VBN, VBZ

Interjections: IN

The NLProcessor system generates XML output. For instance, <W C=‘NN’> indicates a noun and <NG> indicates a noun group/noun phrase. Each sentence is saved in the review database along with the POS tag information of each word in the sentence. A transaction file is then created for the generation of frequent features in the next step. In this file, each line contains words from a sentence, which includes only preprocessed nouns/noun phrases of the sentence. The reason is that other components of a sentence are unlikely to be product features. Here, pre-processing includes the deletion of stopwords, stemming and fuzzy matching. For example,

“autofocus” and “auto-focus” actually refer to the same feature. All the occurrences of “autofocus” are replaced with “auto-focus”.

**4.7 Adjective and Opinion word Extraction:**

Opinion words are words that people use to express a positive or negative opinion. Observing that people often express their opinions of a product feature using opinion words that are located around the feature in the sentence, we can extract opinion words from the review database using all the remaining frequent features (after pruning). For instance, let us look at the following two sentences:

*“The strap is horrible and gets in the way of parts of the camera you need access to.”*

*“After nearly 800 pictures I have found that this camera takes incredible pictures.”*

In the first sentence, *strap*, the feature, is near the opinion word *horrible*. And in the second example, feature *picture* is close to the opinion word *incredible.* Following from this observation, we can extract opinion words in the following way:

• For each sentence in the review database, if it contains any frequent feature, extract the nearby *adjective*. If such an adjective is found, it is considered an opinion word. A nearby adjective refers to the adjacent adjective that modifies the noun/noun phrase that is a frequent feature. As shown in the previous example, *horrible* is the adjective that modifies *strap*, and *incredible* is the adjective that modifies *picture*.

For each opinion word, we need to identify its semantic orientation, which will be used to predict the semantic orientation of each opinion sentence. The semantic orientation of a word indicates the direction that the word deviates from the norm for its semantic group. Words that encode a desirable state (e.g., beautiful, awesome) have a positive orientation, while words that represent undesirable states have a negative orientation After the POS-Tagging Sentences containing adjective (JJ) are kept as important sentences and used as an opinion sentences that will be referred further for summary generation. Thus the presence of adjectives is useful for predicting whether a sentence is subjective, i.e., expressing an opinion.

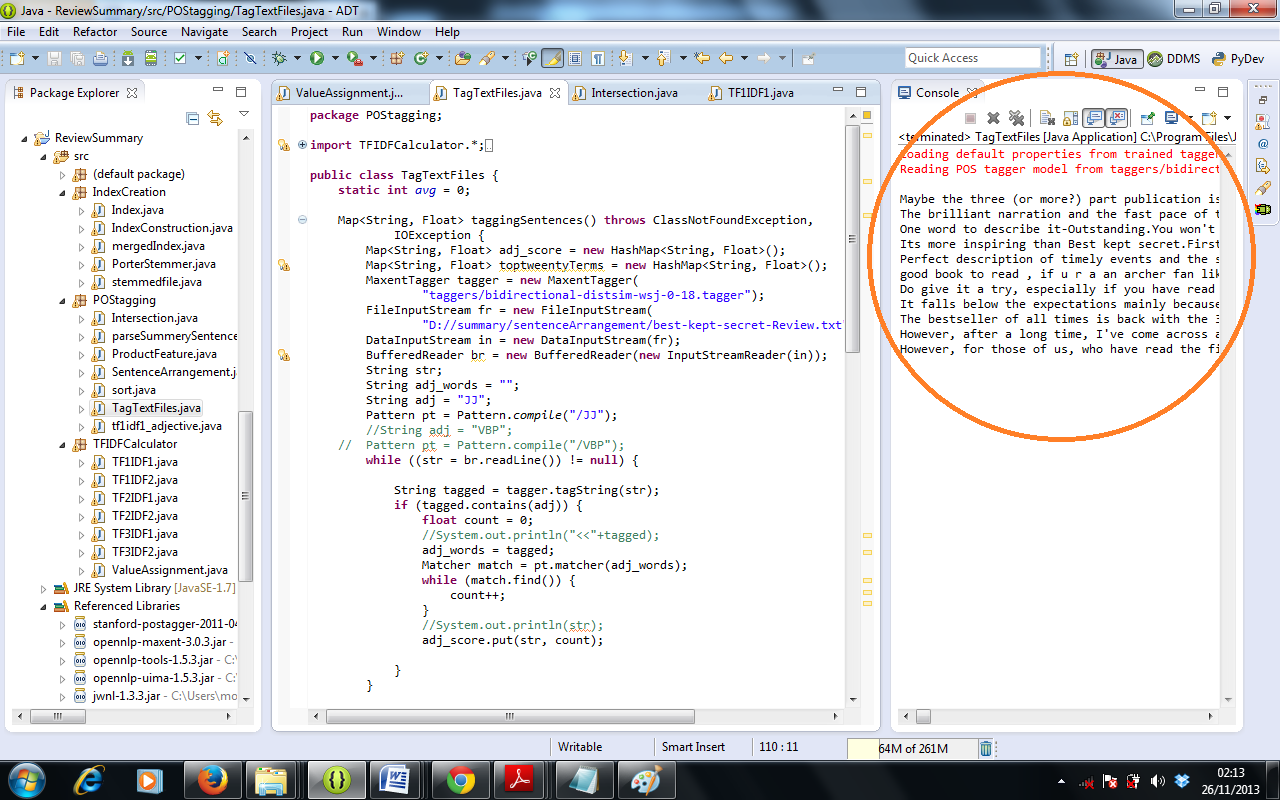
We extract the opinion words in the following manner:

For each sentence in the review database

If (it contains adjective i.e. JJ)

Record the sentence as opinion sentence

The snap short of opinion generated from POS-tagging is as follow:



**4.8 Intersection of both methods:**

The *intersection* of two [sets](http://home.avvanta.com/%7Emath/def2.cgi?t=set) are all the elements that appear in both sets. In our project we use intersection to intersect two different methods for extracting the common opinions from both sets. Let’s say set1 of TF-IDF contains 50 different review sentences and set2 that is generated by analyzing the adjectives contain 40 different opinion sentences. When we intersect both sets the resulting set will contain those sentences that are common in both the sets. Suppose 25 such sentences are there. Then these sentences are shown to user as a summary of that product.

We extract the opinion words in the following manner:

For each sentence in Set1:

For each sentence in Set2:

If(sentence in set1 and set2 is same)

Record that sentence as opinion sentence

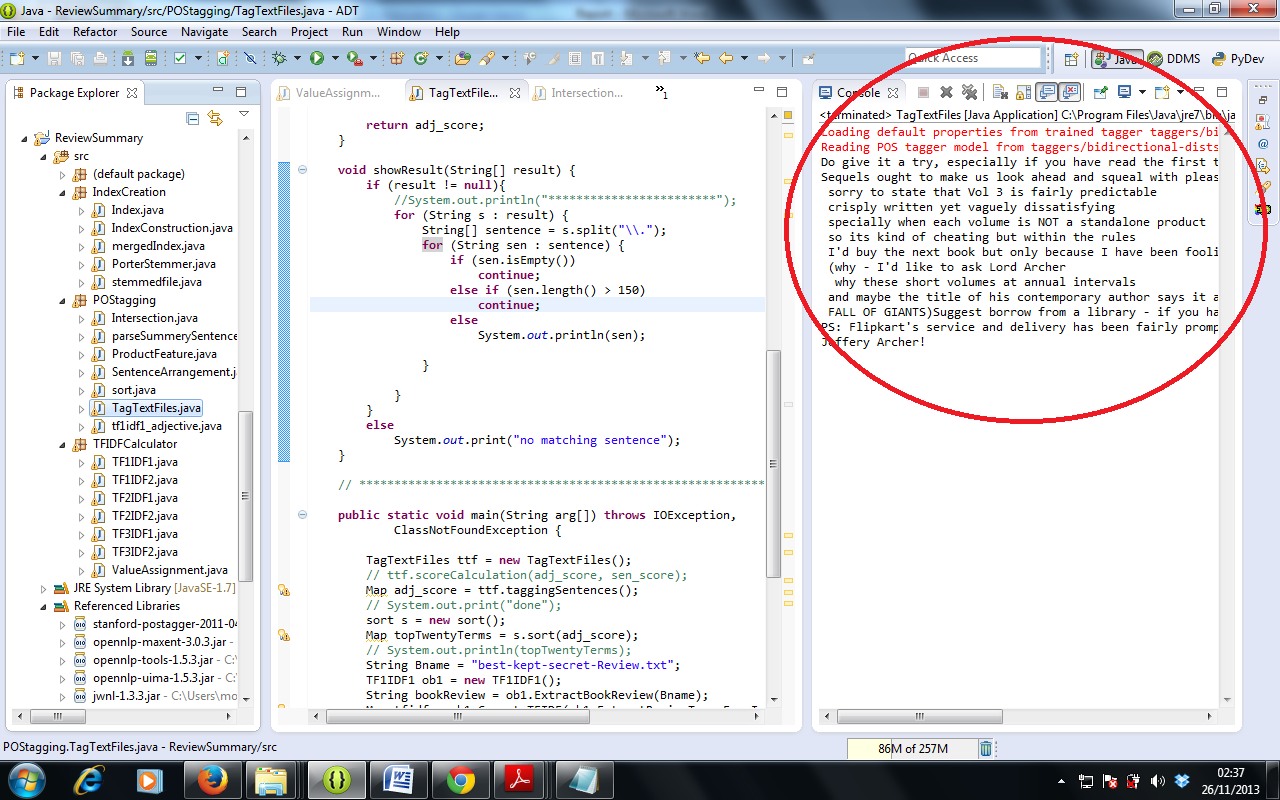


Fig: summary generated by intersecting TF-IDF and POS-Tagging