International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 5 | ISSN : 2456-3307

Categorization of News Articles using Sentiment Analysis

Yashodhara Haribhakta*1, Kiran Shriniwas Doddi 2
*1,2 Department of Computer Engineering and Information Technology, College of Engineering Pune, Pune, Maharashtra, India

ABSTRACT

The advent use of new online social media such as articles, blogs, message boards, news channels, and in general web content has dramatically changed the way people look at various things around them. Today, it's a daily practice for many people to read news online. People's perspective tends to undergo a change as per the news content they read. The majority of the content that we read today is on the negative aspects of various things e .g. corruption, rapes, thefts etc. Reading such news is spreading negativity amongst the people. Positive news seems to have gone into a hiding. The positivity surrounding the good news has been drastically reduced by the number of bad news. This has made a great practical use of Sentiment Analysis and there has been more innovation in this area in recent era. It traditionally emphasizes on classification of text document into positive and negative categories. The objective of this paper is to provide a platform for serving good news and create a positive environment. This is achieved by finding the sentiments of the news articles and filtering out the negative articles which carry negative sentiments. This would enable us to focus only on the good news which will help spread positivity around society and would allow people to think positively. To achieve our objective, we have proposed an algorithm for classification of News articles. This includes data aggregator tool and processing engine at the server side as a Sentiment classifier and a platform for user where positive news being served to read.

Keywords: Document classification, Sentiment Analysis, Support vector machine (SVM)

I. INTRODUCTION

With the arrival of internet, there has been a radical change in the social life, routine and decisions of common people. Today, it's everyday activity and regular practice for each person to read news online and watch advertisements regarding a movie, a product or a book before actually placing money into it. As it has changed their lifestyle, it also has impact on the social life of an individual. The exposure to new online social media such as articles, blogs, message boards, news channels such as Web content is influencing their social life and the way people look at various things around them. People's perspective tends to undergo a change as per the content they read.

The social media has now occupied the major space on the Web. The new user-centric Web hosts a huge amount of data every day. Users are not only consuming the web, but they are also a part of web and co-creators of content on web. The user is now contributing to social media ranging from articles, blog posts, news, tweets, reviews, photo/video upload, etc. This is creating a large amount of the data on the Web as unstructured text.

The majority of the content that we read today is on the negative aspects of various things e.g. corruption, rapes, thefts etc. Reading such news is spreading negativity amongst the people. Positive news has been dominated and getting less attention. The positivity surrounding the good news has been drastically reduced by the number of bad news.

The objective of this project is to provide a platform for serving good news and create a positive environment. The new challenging task here is to analyze large volume of unstructured text to be more specific news articles and devise suitable algorithms to understand the opinion of others and find positive and negative aspect of it. This would enable us to focus only on the good news which will help spread positivity around and would allow people to think positively.

The organization of this paper is as follows. In Section 2, we have discussed the literature survey. Section 3 is design and implementation of the proposed model. Section 4 is experimentation and results. Section 5 is conclusion followed by future directions.

II. LITERATURE SURVEY

In the current situation of the web, we need a way of reducing the amount of effort for processing the unstructured data by transferring it to machine. Computer can process such data in very less amount of time. Only we need to put effort at learning curve of a Computer which is commonly known as Machine Learning. Now days, lot of data is being generated in the form of blogs, reviews, articles etc. Social media has a now a major share on the web. Users are not only consuming the web, but they are also a part of web and co-creators of content on web. The user is now contributing to social media ranging from articles, blog posts, news, tweets, reviews, photo/video upload, etc. This is creating a large volume of the data on the Web as unstructured text . The task here is to analyze the sentiment of such data which is hot research topic in recent era, trending in the market and has a great value.
In previous studies, today researchers have worked on finding sentiments of movie reviews [5], product reviews, twitter messages, prediction on rise or drop in stock price[6] etc. The aspect of such dataset is that it includes short and rich structured information about individuals involved in communication. These data sets often consist of relatively well-formed and coherent pieces of text. Our challenge is to analyze news articles which could be a very large text and sentiment can change from line to line. Finding sentiments of structured data is easier than finding it from unstructured data.

Zi-Jun Yu et al. [1] has written a novel algorithm Keyword combination extraction based on ant colony optimization (KCEACO) to extract most favorable keyword combination from the document which is used for assigning category to the document. They had extended the traditional feature extraction technique to find out the most optimal keyword combinations. Haruechaiyasak, C.[2] has proposed a solution to enhance a search on full text news articles in Thai language. Users were able to browse and filter the search result based on category they select. To implement this feature, they applied and validated several different classification algorithms like Naive Bayes (NB), Decision Tree, and Support Vector machine (SVM). Based on experiment, SVM with information gain as feature selection algorithm had yielded better performance result with F1 measure as 95.42%.

Chua S. [3] proposed a way to find the notion of semantic features using WordNet. This paper tried to find the synonyms of words for finding the feature set which are semantically equivalent to the predefined category of the document. With these efforts, they have found that automated way of categorizing document is much better than any known statistical feature selection method. Feature annotation for text categorization [7] is one of the approaches used for document classification. This paper tries to increase the accuracy of document classification by extracting text collection in four features (dimensions) for each document. The four dimensions were protagonist, temporality, spatiality and organization. The document was annotated with these dimensions and then applied to classifier for classification.

A. Classification Methods and Feature Extraction

For automation of sentiment analysis, different approaches have been invented for predicting sentiment from words, expressions and from documents. There are many natural language processing based and pattern based algorithms like Naïve Bayes (NB), Support Vector Machine(SVM), Maximum Entropy (ME) etc.
However, some research has investigated more complex algorithms. Martineau and Finin invented Delta TFIDF in 2009, an intuitive general purpose technique, so that words can be efficiently weighted before applying to classification [12]. In most of the papers, researchers have used dictionary based document classification technique. Some have used Bag of words technique along with some machine learning algorithms most likely support vector machine. SVM algorithm was mostly used classification algorithm because it is highly generalized and its performance is different for various ranges of applications. It is also considered as one of the most efficient classification algorithm which provides a comprehensive comparison for text classification in supervise machine learning approach.

B. Model Building

Earlier, the model was built once using training data and used to classify the new dataset based on built model. While using so, if new dataset has new features (with the advent of Web and social media), the model fails to classify the document correctly and its accuracy falls. To avoid this, they had to rebuild the new model with new training dataset considering new features and calculate the accuracy. But this is not feasible solution and approach for our problem. News corpus from different sources consists of news articles and editorial content with a broad range of discussions and topics. So challenging task here is to design a generic heuristic algorithm that will correctly extract sentiments from these mixed news corpus for document classification and feature extraction.

C. Data Set

V. K. Singh, R. Piryani, A. Uddin & P. Waila[13] has collected 10 reviews each for 100 Hindi movies from the popular movie review database website www. imdb.com. We have labeled all these reviews manually to evaluate performance of our algorithmic formulations. Out of 1000 movie reviews collected, 760 are labeled positive and 240 are labeled as negative reviews.

Manish Agarwal and Sudeept Sinha[10] has 2000 movie reviews, 1000 positive and 1000 negative. We'll do 5 fold and 10 fold cross validation over the data set. Each fold of the 10-fold will involve 1800 training reviews and 200 test reviews, while for 5-fold cross validation, it will be 1600 training reviews and 400 test reviews. Hitesh Khandelwal[14] has taken 1000 positive and 1000 negative reviews, categorized by them.

III. DESIGN AND IMPLEMENTATION

Analyzing Sentiment of the text is itself a challenging task in Natural Language Processing. There are many approaches studied by me while finding solution for sentiment of news articles. Initially, we collected different words which carry sentiments (like positive, negative and neutral) from DAL, GI and WordNet [3]. Later we needed test data for analyzing the structure of the text . We have read RSS feeds from various sources based on category [4]. With this mixed news corpus on broad range of topics, we want our model to run correctly by extracting new features from dynamic data. So one way to achieve this is, we need to dynamically prepare training dataset after every interval of time, so that system will build the new model with this new corpus at that time and classify correct new test instance. Here is the overall system diagram shown below:

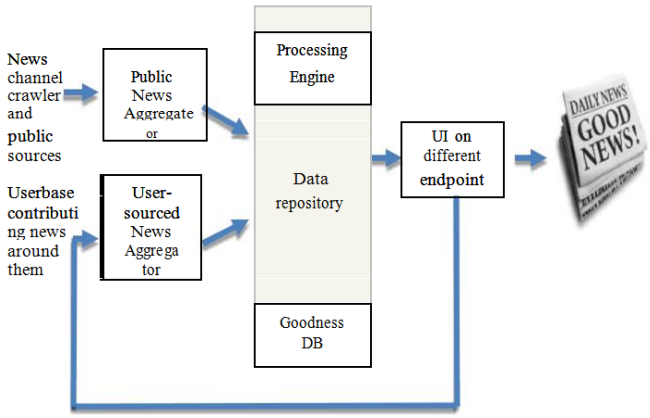

Figure 1. System Design

As shown in figure 1, we have three main components/modules which contributes to form a overall system as below,

1. News Aggregator
2. Processing Engine
3. Platform to serve the good news and publish

We have followed below steps for finding the sentiment of each line of the document as local sentiment L,

1. POS tagger to tag the sentences
2. Extract features (adjectives, adverbs & verbs) and their corresponding scores
3. Construct Term document matrix (Feature extraction) with different weights from SentiWordnet. So the sentiment of overall document (global sentiment) can be calculated as follows:

   $$G = F(L_i)$$

   This global sentiment will define sentiment of overall news article.
4. Change document vector to SVM format
5. Apply SVM to build the model

The built model will be serialized to secondary storage and will be used to classify the new latest articles.

Our system itself is self-explanatory. The most basic objective of our system is to filter out the negative articles from the news corpus. So to accomplish this, several other modules are needed. The solution building blocks are as follows:

A. News Aggregator

RSS is the acronym of Rick Site Summary, is the format to deliver the usually changing web content. Many web blogs, social sites, news related sites and other online publishers syndicate their regularly changing content as RSS feed an d whoever wants to read this can get it in terms of RSS Feeds. Feed reader or news aggregator fetches these feeds from various sites and display them in the format we want. Our system needs RSS feeds from the various news channels. So we have written a module which fetches these feeds from our pre-defined news sources which we have stored in our database, e.g.

Table 1- RSS Feeds List

No.	Feed link	Source
1	http://feeds.feedburner.com/NDTV-LatestNews?format=xml	NDTV
2	http://timesofindia.feedsportal.com/c/33039/f/533916/index.rss	Times Of India
3	http://feeds.feedburner.com/TheBetterIndia?format=xml	The Better India

This RSS feed contains channel's name, description, published date, language under which this feed is published, short description of the channel, its category and a link to their site. It is very difficult to find exact sentiment from those short head line/description. We needed to capture entire news article by visiting its original link, extract article and push it in database. We have mapped this rss xml to its corresponding java class for parsing using JAXB parser.

B. Processing Engine

This is the core module of our project and aim at classifying news article read from the various resources. Our approach is to classify the document into either positive (+) or negative (-) document. There are many types of approaches available for document classification, but broadly three types of approaches are used for document classification for sentiment analysis.

1. Using supervised machine learning algorithm based classifier like Naïve Bayes, Support Vector Machine or kNN with feature extraction scheme
2. Using unsupervised machine learning algorithm for semantic orientation labeling words and then labeling entire document
3. Using publicly available SentiWordNet library which provides positive, negative and neutral score for the words and then label the document based using some heuristic algorithm

News corpus from different sources consists of news articles and editorial content with a broad range of discussions and topics. This also includes huge set of features to be taken into consider. Our aim here is to classify the news articles into positive articles or negative articles. If we see the news channels, they just show the trending news for some period of time and then with new topic people switches to new one completely ignoring older topics.

Let's say, we have built the model and classifier last month considering all the features extracting from the news articles available at that time. But today News articles may contain new features which might not be taken into consideration while building model last month. If we are using same classifier for classification for newly available news articles, then model will wrongly classify it and behave inconsistently. But it is also difficult approach for preparing training data set timely and building the model at some regular interval of time. So to avoid this, why can't we delegate this job to the machine and let it build the model with dynamic data set as some regular interval of time.

To fulfil this approach, we have used SentiWordNet based approach for document classification which involves various linguistic combinations of features (like Adjective + Adverb or Adjective + Adverb and Adverb + Verb combination).

We have explored different approaches for fetching the features from the articles. Computational linguistic says that with some combinations between Adjectives, Adverb and Verb carries more sentiments than any other words in the document and they are the good carriers of sentiments, e.g. “She was allegedly killed”. Here the use of word allegedly tells that killer killed her without any proof. Sometimes, adverb changes the meaning of the sentences, e.g. “He is not terrorist”. Here use of word not which is adverb reversing the overall sentiment of the sentence. So we have tried to classify the documents based on combination of “adverb + adjective” and “adverb + verb” scheme.

In first scheme, we have extracted the adverb and adjective combinations. As adverbs modifies the sentiment of the succeeding adjective or adverb, we have to decide as in what proportion or scale it should modify the sentiscore of succeeding adjective or verb. T his is needed to achieve the highest accuracy. So we have tried with various values for this factor ranging from 10% to 100%. And from our analysis and dataset, to get higher accuracy, we have taken this scaling factor sf = 0.35.

Below is the pseudo code for the adverb + adjective combination scheme with scaling factor sf = 0.35.

1. For each sentence, extract Adverb + Adjective combination

2. totalAdvAdjCount := 0

3. For each extracted (Adv + Adj) combination do:

- If sentiscore(Adv) > 0,
  - If sentiscore(Adj) is positive then,
    sentenceScore += sentiscore(Adj) + sf*sentiscore (Adv)
  - If sentiscore(Adj) is negative then,
    sentenceScore += sentiscore(Adj) - sf*sentiscore(Adv)
  - Increment total no of AdjAdv count (totalAdvAdjCount)

- If sentiscore(Adv) < 0,
  - If sentiscore(Adj) is positive then,
    sentenceScore += sentiscore(Adj) + sf*sentiscore(Adv)
  - If sentiscore(Adj) is negative then,
    sentenceScore += sentiscore(Adj) - sf*sentiscore(Adv)
  - Increment total no. of AdvAdj count (totalAdvAdjCount)

- If sentiscore(Adj) == 0, ignore it
- If sentiscore(Adv) == 0, ignore it

4. FinalSentenceScore(Adv,Adj) = sentenceScore/totalAdvAdjCount;

Here Adj is acronym for adjectives, Adv is acronym for adverb, sentiscore is acronym for sentiment score from SentiWordNet. Here we processed all the sentences and final document score is aggregated from the FinalSentenceScore of each sentence.

In second scheme, we have tried to combine both the combinations “Adverb + Adjective” and “Adverb + Verb”. This scheme is similar to the previous one in combining Adverb with Adjective. Here also we have used same scaling factor used in first scheme i.e., sf=0.35.

Below is the pseudo code for the adverb + adjective and adverb + verb combination scheme with scaling factor sf=0.35.

1. For each sentence, extract Adverb + Adjective combination
2. totalAdvAdjCount := 0
3. For each extracted (Adv + Adj) combination do:
4. For each extracted (Adv + Adj) combination do:

- If sentiscore(Adv) > 0,
  - If sentiscore(Adj) is positive then,
    sentenceScore += sentiscore(Adj) + sf*sentiscore (Adv)
  - If sentiscore(Adj) is negative then,
    sentenceScore += sentiscore(Adj) - sf*sentiscore(Adv)
  - Increment total no of AdjAdv count (totalAdvAdjCount)
- If sentiscore(Adv) < 0,
  - If sentiscore(Adj) is positive then,
    sentenceScore += sentiscore(Adj) + sf*sentiscore(Adv)
  - If sentiscore(Adj) is negative then,
    sentenceScore += sentiscore(Adj) - sf*sentiscore(Adv)
  - Increment total no. of AdvAdj count

(totalAdvAdjCount)

- If sentiscore(Adj) == 0, ignore it
- If sentiscore(Adv) == 0, ignore it

4. FinalSentenceScore(Adv,Adj) = sentenceScore/totalAdvAdjCount;
5. For each sentence, extract Adverb + Verb combination
6. totalAdvVerbCount := 0
   For each extracted (Adv + Verb) combination do:

- If sentiscore(Adv) > 0,
  - If sentiscore(Verb) is positive then, sentenceScore += sentiscore(Verb) + sf*sentiscore(Adv)
  - If sentiscore(Verb) is negative then, sentenceScore += sentiscore(Verb) - sf*sentiscore(Adv)
  - Increment total no of AdjAdv count (totalAdvAdjCount)
- If sentiscore(Adv) < 0,
  - If sentiscore(Verb) is positive then, sentenceScore += sentiscore(Verb) + sf*sentiscore(Adv)
  - If sentiscore(Verb) is negative then, sentenceScore += sentiscore(Verb) - sf*sentiscore(Adv)
  - Increment total no. of Adv Verb count (totalAdvVerb Count)
- If sentiscore(Adv) == 0, ignore it
- If sentiscore(Verb) == 0, ignore it

7. FinalSentenceScore(Adv, Verb) = sentenceScore/totalAdvVerb Count;
8. FinalScore(Sentence) = FinalSentenceScore(Adv, Adj) + 0.6 * FinalSentenceScore(Adv, Verb)

Here, we processed all the sentences first and then final document score is aggregated from the Final Sentence Score of each sentence.

In the process of preparing training data set, we are also extracting different attributes as a feature listed below,

1. # positive words
2. # negative words
3. # neutral words
4. # strong negative words

After getting training data set ready from above algorithms, we are adding the features mentioned above and then converting training set into CSV file. Now this CSV contains features of each document of training data set. Now next step is to convert this CSV file into a format which our machine algorithm understands. As we have used WEKA libraries, we need to convert CSV to ARFF file format. Weka itself provides APIs to convert CSV file directly into ARFF file format. CSVLoader class takes csv file as input which needs features to be in comma separated list. Later ArffSaver is one of the classes in Weka which takes care of converting csv into arff file. We need to set instances to ArffSaver class and this class writes in arff file format .

After our training data set is ready in the arff format, we are preparing out test data in the same way mentioned above for training data. In Weka, It is necessary for machine learning algorithms to make instances available in arff file format. Now we have arff file for training data set as well as for test data set and we will use them as Data source to pass them to Weka. We have used SVM algorithm for building our model as classifier.

We have experimented our data set with different kernels. We have used PolyKernel, Gaussian and String Kernel with different values of exponent, gamma and lambda respectively.

We have taken total 480 news articles, out of which 265 were positive and 215 were negative. Using above algorithm, we extracted 428 unique pairs of adverb + adjective combination and 1169 unique pairs of adverb + verb combinations.

C. Platform to Serve the Positive News

This is our end point where users log in and read the positive news. News can be picked up from repository which are processed and is assigned positive label. News is shown to user in the same format how he can actually read it on original site. Today we have built a Web UI to read the positive news.

To build this platform, we have used divided this into two sub-modules.

1. Backend REST APIs
2. Web UI

D. Data Set

For our analysis, we have gathered news articles from different RSS feeds mentioned at [15]. We have pre-

defined some news channels in the database and fetched around more than 5000 news articles for analysis. Out of which we are considering only 500 articles as a training data set .

This includes positive as well as negative news articles. We have taken news articles of different categories like Gadgets, India, World, Sports, Cities, Bollywood, and Economy etc.

We have also collected sentiment words from GI, DAL and WordNet [12].

E.g.: Positive words: awesome, special, active, sprite, strong, moral, beautiful, visionary, attractive etc.

Negative words: abused, hate, accused, awkward, betrayed, crap, dark, dead, defective, sad, screwed etc.

Neutral words: absolute, alert, assess, emotion, engage, entire, fact, feel, foresee, imagine, idea, infer etc.

IV. EXPERIMENTS AND RESULTS

In this section, we present the experiments carried out with our data set and the end result of it. Here, we discuss our heuristic algorithm for document classification and SVM with different Kernel by setting different values to its corresponding attribute.

A. Dataset Used for Experimentation

We have designed our own cronjob (Data-Aggregator) which runs every hour, fetches data from feedburner (RSS server for news articles) and store it in MySQL database. We have collected around more than 25000 articles. Out of them, around 480 articles are prepared for our experiment purpose. We have 480 news articles preclassified with positive and negative labels. Out of them, 265 articles are positive and 215 articles are negative. We have also prepared test data set of 50 articles out of which 25 are positive and 25 are negative. We also have dictionary of positive, negative and neutral words by using which we calculate the feature vector.

B. Measures Used for Evaluation

For classification, we have using confusion matrix (contingency table) of four different measures, true positive, true negative, false positive and false negative. Our system is trained to calculate these four measures and we are calculating other important measures using them. This is illustrated in Table 2:

Table 2: Confusion matrix for Classification Model

	Actual label (Expectation)	
Predicted label (Observation)	TP (True Positive) Correct result	FP (False Positive) Unexpected result
	FN (False Negative) Missing result	TN (True Negative) Correct result

Precision: % of selected documents that are correctly classified in class C out of all documents in class C.
Recall: % of correct documents that are selected in class C from all the documents actually belonging to class C.
Accuracy: Proportion of total number of predictions that are correctly classified in class C.

C. Result of Document Classification

We have mentioned our heuristic algorithm for document classification in section III. In this section, we are presenting the results for different values of adverb scaling factor (sf) which takes the % of adverb share in calculating sentiment of a line and verb scaling factor which takes % of verb share in calculating sentiment of document .

Table 3: Adverb-verb scaling factors

Adverb Scaling factor	0.25	0.35	0.35	0.45	0.55	0.55
Verb Scaling factor	0.2	0.3	0.4	0.4	0.4	0.5
Accuracy	0.8552	0.8574	0.8552	0.8531	0.8509	0.848
Precision	0.9159	0.9128	0.9090	0.9087	0.9117	0.908
Recall	0.8226	0.8301	0.8301	0.8264	0.8188	0.818

As shown in table 3, we are getting maximum accuracy for Adverb Scaling Factor 0.35 and Verb Scaling Factor 0.3. So we have used these results to compute and validate further result by SVM with

different Kernels and with their corresponding attribute values.

D. Result of SVM algorithm with different Kernels

Here we have taken Adverb Scaling factor = 0.35 and Verb Scaling factor as 0.3. First we have set Gaussian Kernel in SVM and started its gamma value from 1.1 till 2.0. Corresponding class name for Gaussian Kernel in Weka is RBFKernel.

Table 4: SVM with Gaussian Kernel

Kernel	RBFKernel	RBFKernel	RBFKernel	RBFKernel
Gamma	1.1	1.3	1.5	1.8
Accuracy	0.6086	0.6521	0.6956	0.7173
Precision	0.6071	0.6071	0.6071	0.6071
Recall	0.7083	0.7727	0.85	0.8947

From table 4, we found that for Gamma value of 1.8, we are getting maximum accuracy, precision and recall. So we achieve an Accuracy=0.717391304, Precision=0.607142857 and Recall=0.894736842.

Later we have verified result for Polynomial Kernel in SVM and started its exponent value from 2 till 5. Corresponding class name for Gaussian Kernel in Weka is PolyKernel.

Table 5 shows the result for different exponent values of Polynomial Kernel.

Table 5 : SVM with Polynomial Kernel(PK)

Kernel	PK	PK	PK	PK
Exponent	2	3	4	5
Accuracy	0.6956	0.6739	0.6521	0.6521
Precision	0.8214	0.8571	0.8571	0.8571
Recall	0.7187	0.6857	0.6667	0.6667

Here we found that for Exponent value of 2, we are getting maximum accuracy, precision and recall. So Accuracy=0.695652174, Precision=0.821428571 and Recall=0.71875.

V. CONCLUSIONS

The proposed solution and implementation of the system justify that we can build our model on dynamic data set of news corpus on broad range of topics and we are getting satisfactory performance for document classification for preparing training data set which is very crucial step of classification.

We can use this proposed system in the area where feature of documents changes as per time and preparing training data again for new features is important .

VI. REFERENCES

[1] Zi-jun Yu, Wei-gang Wu, Jin g Xiao, Jun Zhang, Rui-Zhang Huang, OuLiu, "Keyword Combination Extraction in Text Categorization Based on Ant Colony Optimization" in International Conference of Soft Computing and Pattern Recognition, 2009. SOCPAR '09, pp.430-435, 4-7 Dec. 2009.

[2] Choochart Haruechaiyasak, Wittawat Jitkrittum, Chatchawal Sangkeettrakarn, Chaianun Damrongrat, "Implementing News Article Category Browsing Based on Text Categorization Technique", in IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08.

[3] Stephanie Chua, Narayanan Kulathuramaiyer, "Semantic Feature Selection Using WordNet" in IEEE/WIC/ACM International Conference on Web Intelligence, Pages 166-172, September 20-24, Pgs.166-172, Sept. 20-24, 2004.

[4] Yi Guo, Zhiqing Shao and Hua Nan, "Content-Oriented Automatic Text Categorization with the Cognitive Situation Models", International Symposium on Computer Science and Computational Technology, in 2008.

[5] A new Feature-based Heuristic for Aspect-level Sentiment Classification by V.K. Singh, R. Piryani, A. Uddin & P. Waila.

[6] Sentiment Classification for Stock News.

[7] Wikipedia-http://en.wikipedia.org/wiki/ Unstructured_data

[8] Y.V. Haribhakta, Santosh Kalamkar, Dr.Parag Kulkarni , "Feature annotation for text categorization", CUBE International Information Technology Conference, ACM ICPS Proceedings, September 3-5, 2012, Pune, India.

[9] Fabrizio Sebastiani, “Machine learning in automated text categorization” in ACM Computer Survey 34,March 2002.
[10] Polarity Detection in Reviews(Sentiment Analysis) by Manish Agarwal and Sudeept Sinha, 2009-10, IIT-Kanpur, Report.
[11] POS Tagger (The Stanford Natural Language Processing Group)
[12] WordNet: An Electronic Lexical Database http://wordnet.princeton.edu .
[13] Sentiment Analysis of movie review by V.K. Singh, R. Piryani, A. Uddin & P. Waila .
[14] Polarity detection in movie reviews by Hitesh Khandelwal (Y5202) .
[15] Fetch RSS feeds from feedburner.com and feedporter.com.