International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 7 | ISSN : 2456-3307

A Survey On Opinion Mining and Sentiment Analysis Using R - Programming

Sravani S
Assistant Professor, Department of Computer Applications, K.B.N College, P.G Centre, Vijayawada, Andhra Pradesh, India

ABSTRACT

An important part of our information-gathering behavior has always been to find out what other people think. With the growing availability and popularity of opinion-rich resources such as online review sites and personal blogs, new opportunities and challenges arise as people now can, and do, actively use information technologies to seek out and understand the opinions of others. Sentiment analysis (also known as opinion mining) refers to the use of natural language processing (NLP), text analysis and computational linguistics to identify and extract subjective information from the source materials .The sudden eruption of activity in the area of opinion mining and sentiment analysis, which deals with the computational treatment of opinion, sentiment, and subjectivity in text, has thus occurred at least in part as a direct response to the surge of interest in new systems that deal directly with opinions as a first-class object. Generally speaking, sentiment analysis aims to determine the attitude of a writer or a speaker with respect to a specific topic or the overall contextual polarity of a document. Globally, business enterprises can leverage opinion polarity and sentiment topic recognition to gain deeper understanding of the drivers and the overall scope. Subsequently, these insights can advance competitive intelligence and improve customer service, thereby creating a better brand image and providing a competitive edge.

Keywords : Natural Language Processing, Opinion Mining

I. INTRODUCTION

The e-commerce industry is benefitting greatly by utilizing sentiment analysis. Generally, on e-commerce portals, buyers often express their opinions in the form of comments (positive or negative) for the products they have purchased, making this a huge data trove for sentiment analysis. Correspondingly, analysis of such opinion-related data (comments) can provide deep-insights to the key stakeholders. A thorough sentiment analysis reveals deep-insights on the product, quality and performance. Additional insights that can be extracted using sentiment analysis include.

- Insights on competitors
- Feedback on newly launched products
- Influencing factors affecting other customer decisions
- Company news and trends

Given our mandate, the reader will not be surprised that we describe the applications that sentiment-analysis systems can facilitate and review many kinds of approaches to a variety of opinion-oriented classification problems.We have also chosen to attempt to draw attention to single- and multi-document summarization of evaluative text, especially since interesting

considerations regarding graphical visualization arise. Finally, we move beyond just the technical issues, devoting significant attention to the broader implications that the development of opinion-oriented information-access services have: we look at questions of privacy, manipulation, and whether or not reviews can have measurable economic impact.

Sentiment analysis is widely used across the financial domain for trading and investing. Notably, financial analysts and traders monitor/analyze social networks to quickly identify the trending stocks and fluctuations in the stock markets, which enable them to react swiftly to any major changes in the stock market. The interest that individual users show in online opinions about products and services, and the potential influence such opinions wield, is something that vendors of these items are paying more and more attention

II. OBJECTIVES OF THE STUDY:

- The approach followed here is to count the positive and negative words in each tweet and assign a sentiment score. This way, we can ascertain how positive or negative a tweet is.
- TwitterR offers an easy way to extract tweets containing a given hashtag, word or term from a user's account or public tweets.

SENTIMENT ANALYSIS APPROACH:

The approach followed here is to count the positive and negative words in each tweet and assign a sentiment score. This way, we can ascertain how positive or negative a tweet is. Nevertheless, there are multiple ways to calculate such scores; here is one formula to perform such calculations.

Score = Number of positive words - Number of negative words

If Score > 0, means that the tweet has 'positive sentiment'

If Score < 0, means that the tweet has 'negative sentiment'

If Score = 0, means that the tweet has 'neutral sentiment'

To find out the list of positive and negative words, an opinion lexicon (English language) can be utilized.

Extracting and Analyzing Tweets

TwitterR offers an easy way to extract tweets containing a given hashtag, word or term from a user's account or public tweets. However, before loading twitterR library and using its functions, developers should create an app on dev.twitter.com and then run the following code, which is written in the R programming language.

Setting Authorization to Extract Tweets

Run the following code in the R Studio to set authorization to extract tweets.

```
reqURL <-
"https://api.twitter.com/oauth/request_token"
accessURL <-
"http://api.twitter.com/oauth/access_token"
authURL <- "http://api.twitter.com/oauth/authorize"
api_key <- "yourconsumerkey"
api_secret <- "yourconsumersecret"
access_token <- "consumeraccess token"
access_token_secret <- "consumer access secret
token"
setup_twitter_oauth(api_key,api_secret,access_token,
access_token_secret)
```

Required Libraries

Here is the code to load required libraries.

```
library(twitteR) ### for fetching the tweets
library(plyr) ## for breaking the data into
manageable pieces
library(ROAuth) # for R authentication
library(stringr) # for string processing
library(ggplot2) # for plotting the results
```

Importing Files

Developers have to import files containing a dictionary of positive and negative words. Likewise, text files containing positive and negative sentiments can be imported using the below code. These files can be downloaded using the Google search engine.

```
posText <- read.delim("…/positive-words.txt",
header=FALSE, stringsAsFactors=FALSE)
 posText <- posText$V1
 posText <- unlist(lapply(posText, function(x)
{ str_split(x, "\n") }))
 negText <- read.delim("…/negative-words.txt",
header=FALSE, stringsAsFactors=FALSE)
 negText <- negText$V1
 negText <- unlist(lapply(negText, function(x)
{ str_split(x, "\n") }))
 pos.words = c(posText, 'upgrade')
 neg.words = c(negText, 'wtf', 'wait',
'waiting','epicfail', 'mechanical')
```

NTERPRETATIONS

Extracting Tweets with Hashtags

To demonstrate sentiment analysis, we analyzed tweets relating to Delta, JetBlue and United Airlines. In order to extract specific tweets relating to these airlines, developers should query twitter for tweets with the hashtag Delta, JetBlue and United.

```
delta_tweets = searchTwitter('@delta', n=5000)
jetblue_tweets = searchTwitter('@jetblue', n=5000)
united_tweets = searchTwitter('@united', n=5000)
```

Processing Tweets

Step 1 – Convert the tweets to a text format.

```
delta_txt = sapply(delta_tweets, function(t)
t$getText() )
jetblue_txt = sapply(jetblue_tweets, function(t)
t$getText() )
united_txt = sapply(united_tweets, function(t)
t$getText() )
```

Step 2 – Calculate the number of tweets for each airline.

```
noof_tweets = c(length(delta_txt),
length(jetblue_txt),length(united_txt))
```

Step 3 – Combine the text of all these airlines

```
airline<- c(delta_txt,jetblue_txt,united_txt)
```

Sentiment Analysis Application (Code)

The code below showcases how sentiment analysis is written and executed. However, before we proceed with sentiment analysis, a function needs to be defined that will calculate the sentiment score.

```
score.sentiment = function(sentences, pos.words,
neg.words, .progress='none'{
# Parameters
# sentences: vector of text to score
# pos.words: vector of words of positive sentiment
# neg.words: vector of words of negative sentiment
# .progress: passed to laply() to control of progress
bar
# create a simple array of scores with laply
scores = laply(sentences,
function(sentence, pos.words, neg.words)
{
# remove punctuation
sentence = gsub("[[:punct:]]", "", sentence)
# remove control characters
```

```
sentence = gsub("[[:cntrl:]]", "", sentence)
# remove digits?
sentence = gsub('\\d+', '', sentence)
# define error handling function when trying
tolower
tryTolower = function(x)
{
# create missing value
y = NA
# tryCatch error
try_error = tryCatch(tolower(x), error=function(e)
e)
# if not an error
if (!inherits(try_error, "error"))
y = tolower(x)
# result
return(y)
}
# use tryTolower with sapply
sentence = sapply(sentence, tryTolower)
# split sentence into words with str_split (stringr
package)
word.list = str_split(sentence, "\\s+")
words = unlist(word.list)
# compare words to the dictionaries of positive &
negative terms
pos.matches = match(words, pos.words)
neg.matches = match(words, neg.words)
# get the position of the matched term or NA
# we just want a TRUE/FALSE
pos.matches = !is.na(pos.matches)
neg.matches = !is.na(neg.matches)
# final score
score = sum(pos.matches) - sum(neg.matches)
return(score)
}, pos.words, neg.words, .progress=.progress )
# data frame with scores for each sentence
scores.df = data.frame(text=sentences, score=scores)
return(scores.df)
}
```

Now, we can start processing the tweets to calculate the sentiment score.

```
1  scores = score.sentiment(airline,
   pos.words,neg.words , .progress='text')
```

Step 1 – Create a variable in the data frame.

```
1  scores$airline = factor(rep(c("Delta",
   "JetBlue","United"), noof_tweets))
```

Step 2 – Calculate positive, negative and neutral sentiments.

```
1 scores$positive <- as.numeric(scores$score >0)
2 scores$negative <- as.numeric(scores$score >0)
3 scores$neutral <- as.numeric(scores$score==0)
```

Step 3 – Split the data frame into individual datasets for each airline.

```
1  delta_airline <- subset(scores,
   scores$airline=="Delta")
2  jetblue_airline <-
   subset(scores,scores$airline=="JetBlue")
3  united_airline <-
   subset(scores,scores$airline=="United")
```

Step 4 – Create polarity variable for each data frame.

```
delta_airline$polarity <-
ifelse(delta_airline$score   >0,"positive",ifelse(delta_a
1irline$score <
0,"negative",ifelse(delta_airline$score==0,"Neutral",0
)))

jetblue_airline$polarity <-
ifelse(jetblue_airline$score >0,"positive",ifelse(jetblu
le_airline$score <
0,"negative",ifelse(jetblue_airline$score==0,"Neutral"
,0)))
```

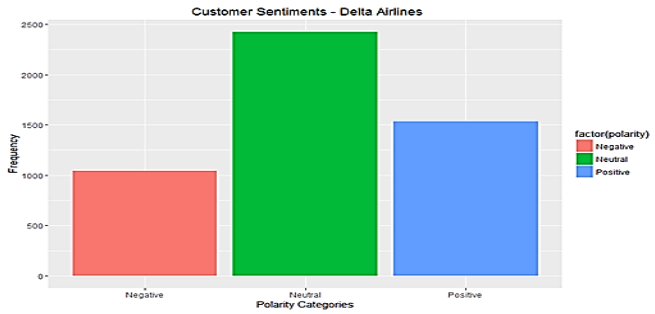
```
united_airline$polarity <-
ifelse(united_airline$score >0,"positive",ifelse(united
1_airline$score <
0,"negative",ifelse(united_airline$score==0,"Neutral"
,0)))
```

Generating Graphs

After the above steps are executed, developers can go ahead and create insightful graphs. The steps below outline the process to create graphs.

Polarity Plot – Customer Sentiments (Delta Airlines)

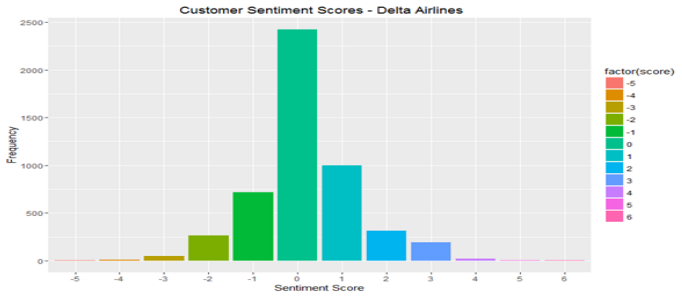
```
1 qplot(factor(polarity), data=delta_airline,
geom="bar", fill=factor(polarity))+xlab("Polarity
Categories") + ylab("Frequency") + ggtitle("Customer
Sentiments - Delta Airlines")
```



The bar graph above depicts polarity, if we closely analyze the graph; it reveals that out of 5,000 twitter users, 1,100 twitter users have commented in a negative way, 2,380 users are neutral. However, 1,520 users are pretty positive about the airline.

```
qplot(factor(score), data=delta_airline, geom="bar",
1 fill=factor(score))+xlab("Sentiment Score") +
ylab("Frequency") + ggtitle("Customer Sentiment
Scores - Delta Airlines")
```

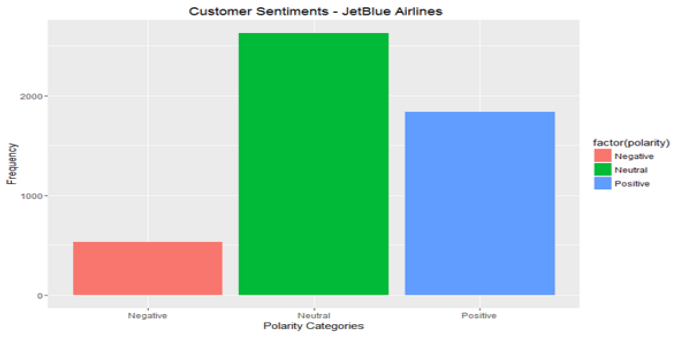
Customer Sentiment Scores (Delta Airlines)



The bar graph above depicts twitter user's sentiment score, negative score denoted by the (-) symbol, which indicates unhappiness of users with the airline, whereas the positive score denotes that users are happy with the airline. While, zero represents that twitter users are neutral.

Polarity Plot – Customer Sentiments (JetBlue Airlines)

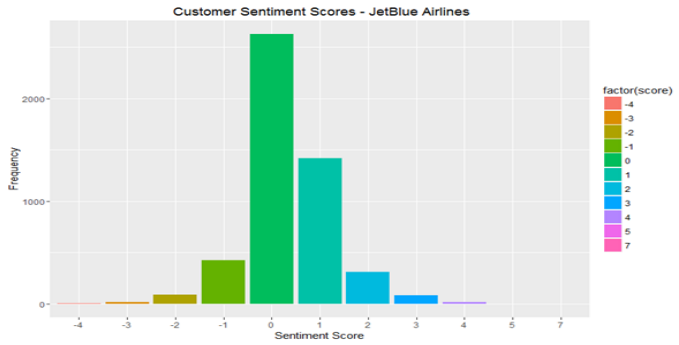
```
qplot(factor(polarity), data=jetblue_airline,
1 geom="bar", fill=factor(polarity))+xlab("Polarity
Categories") + ylab("Frequency") + ggtitle("
Customer Sentiments - JetBlue Airlines ")
```



The bar graph above represents polarity. In this case, out of the 5,000 twitter users, 550 users have commented negatively, 2,700 users remain neutral, whereas 1,750 users are positive about the airline.

Customer Sentiment Scores (JetBlue Airlines)

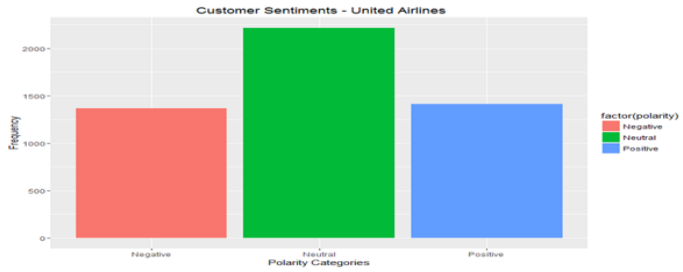
```
1 qplot(factor(score), data=jetblue_airline,
geom="bar", fill=factor(score))+xlab("Sentiment
Score") + ylab("Frequency") + ggtitle("Customer
Sentiment Scores - JetBlue Airlines")
```



The bar graph above depicts twitter user's sentiment score, negative score denoted by the (-) symbol, which indicates unhappiness with the airline, whereas the positive score denotes that users are quite happy. Whereas, zero here represents that users are neutral.

Polarity Plot – Customer Sentiments (United Airlines)

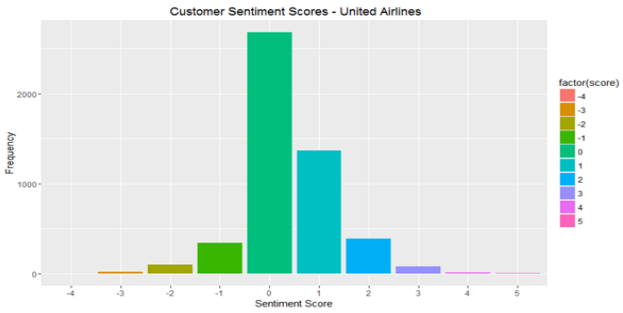
```
qplot(factor(polarity), data=united_airline,
geom="bar", fill=factor(polarity))+xlab("Polarity
Categories") + ylab("Frequency") + ggtitle("Customer
Sentiments - United Airlines")
```



The bar graph above represents polarity. In this case, out of the 5,000 twitter users, 1,350 users have commented negatively, whereas 2,200 users are neutral and remaining 1,450 users remain positive about the airline.

Customer Sentiment Scores (United Airlines)

```
qplot(factor(score), data=united_airline, geom="bar",
fill=factor(score))+xlab("Sentiment Score") +
ylab("Frequency") + ggtitle("Customer Sentiment
Scores - United Airlines ")
```



The bar graph above depicts twitter user's sentiment score, negative score denoted by the (-) symbol indicates unhappiness of users with the airline, whereas the positive score denotes that users are quite happy. While, zero represents that users are neutral about their opinion.

Summarizing Scores

- The code below will help developers to summarize the overall positive, negative and neutral scores

```
df = ddply(scores, c("airline"), summarise,
pos_count=sum( positive ),
neg_count=sum( negative ),
neu_count=sum(neutral))
```

- To put it in another way, developers can create total count by adding positive, negative and neutral sum.

```
df$total_count = df$pos_count
+df$neg_count + df$neu_count
```

- Additionally, developers can calculate positive, negative and neutral percentages using the below code.

```
df$pos_prcnt_score = round( 100 *
df$pos_count / df$total_count )
df$neg_prcnt_score = round( 100 *
df$neg_count / df$total_count )
df$neu_prcnt_score = round( 100 *
df$neu_count / df$total_count )
```

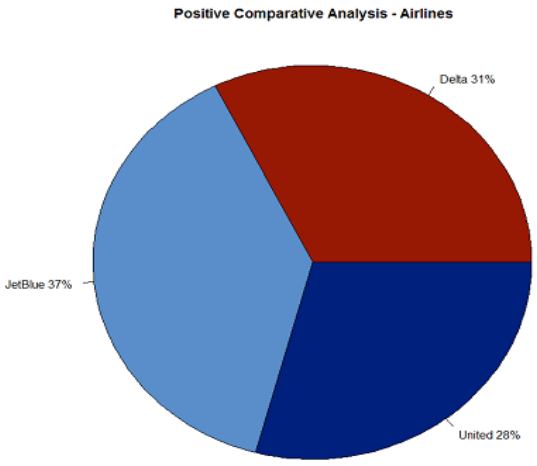
Comparison Charts

Positive Comparative Analysis

Here is the code to create a positive comparison pie chart for these three airlines:

```
attach(df)
lbls <-paste(df$airline,df$pos_prcnt_score)
lbls <- paste(lbls,"%",sep="")
pie(pos_prcnt_score, labels = lbls, col =
rainbow(length(lbls)), main = "Positive Comparative
Analysis - Airlines")
```

The pie chart below represents positive percentage score of these airlines.

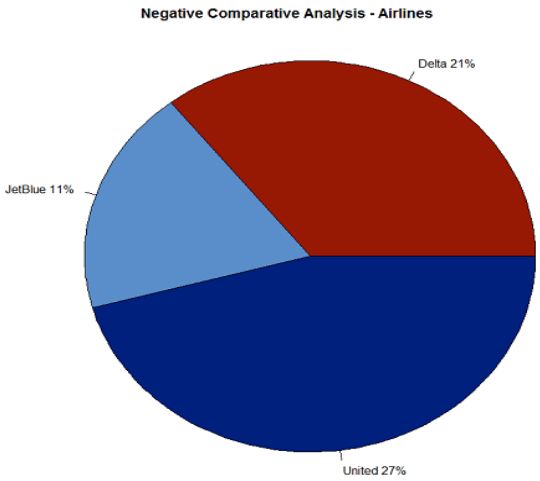


Negative Comparative Analysis

Here is the code to create a negative comparison pie chart for these three airlines:

```
lbls <-paste(df$airline,df$neg_prcnt_score)
lbls <- paste(lbls,"%",sep="")
pie(neg_prcnt_score, labels = lbls, col =
rainbow(length(lbls)), main = " Negative Comparative
Analysis - Airlines")
```

The pie chart below represents negative percentage score of these three airlines.

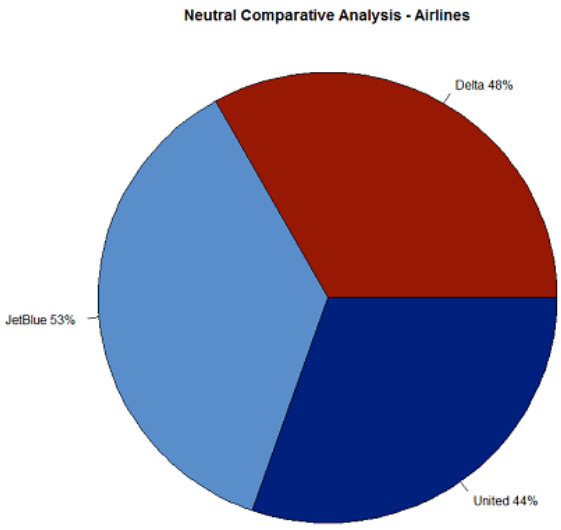


Neutral Comparative Analysis

Here is the code to create a neutral comparison pie chart:

```
lbls <-paste(df$airline,df$neu_prcnt_score)
lbls <- paste(lbls,"%",sep="")
pie(neu_prcnt_score, labels = lbls, col =
rainbow(length(lbls)), main = "Neutral Comparative
Analysis - Airlines")
```

The pie chart below represents neutral percentage score of these three airlines.



III. CONCLUSION

As can be seen, sentiment analysis enables enterprises to understand consumer sentiments in relation to specific products/services. Moreover, these insights could be used to improve their products and services by gauging consumers' comments and feedback using sentiment analysis. In the long run, sentiment analysis, if implemented the right way can aid business enterprises in improving the overall consumer experience, enhance brand image and propel business growth. A sizeable number of papers mentioning "sentiment analysis" focus on the specific application of classifying reviews as to their polarity (either positive or negative), a fact that appears to have caused some authors to suggest that the phrase refers specifically to this narrowly defined task. However, nowa- days many construe the term more broadly to mean the computational treatment of opinion, sentiment, and subjectivity in text.

Thus, when broad interpretations are applied, "sentiment analysis" and "opinion mining" denote the same field of study (which itself can be considered a sub-area of subjectivity analysis). We have attempted to use these terms more or less interchangeably in this survey.

IV. REFERENCES

1. R K. Ando and T. Zhang, "A framework for learning predictive struc- tures from multiple tasks and unlabeled data," Journal of Machine Learning Research, vol. 6, pp. 1817-1853, 2005.
2. A Andreevskaia and S. Bergler, "Mining WordNet for a fuzzy sentiment: Sen- timent tag extraction from WordNet glosses," in Proceedings of the European Chapter of the Association for Computational Linguistics (EACL), 2006.
3. W Antweiler and M. Z. Frank, "Is all that talk just noise? The informa- tion content of internet stock message boards," Journal of Finance, vol. 59, pp. 1259-1294, 2004.
4. N Archak, A. Ghose, and P. Ipeirotis, "Show me the money! Deriving the pricing power of product features by mining consumer reviews," in Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2007.
5. S Argamon, ed., Proceedings of the IJCAI Workshop on DOING IT WITH STYLE: Computational Approaches to Style Analysis and Synthesis. 2003.
6. S Argamon, J. Karlgren, and J. G. Shanahan, eds., Proceedings of the SIGIR Workshop on Stylistic Analysis of Text For Information Access. ACM, 2005.
7. S Argamon, J. Karlgren, and O. Uzuner, eds., Proceedings of the SIGIR Work- shop on Stylistics for Text Retrieval in Practice. ACM, 2006.
8. Tsur, D. Davidov, and A. Rappoport, " A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews". In Proceeding of ICWSM.of Context Dependent Opinions,2010.
9. Weitong Huang, Yu Zhao, Shiqiang Yang, Yuchang Lu, "Analysis of the user behavior and opinion classification based on the BBS" , Applied Mathematics and Computation 205 (2008) 668-676 .
10. Wiebe, W.-H. Lin, T. Wilson and A. Hauptmann, "Which side are you on? Identifying perspectives at the document and sentence levels," in Proceedings of the Conference on Natural Language Learning (CoNLL), 2006.
11. Yi and Niblack, "Sentiment Mining in Web Fountain",Proceedings of 21st international Conference on Data Engineering, pp. 1073-1083, Washington DC,2005.

12. Yongyong Zhail, Yanxiang Chenl, Xuegang Hu, "Extracting Opinion Features in Sentiment Patterns" ,International Conference on Information, Networking and Automation (ICINA),2010.
13. YuanbinWu, Qi Zhang, Xuanjing Huang, LideWu, "Phrase Dependency Parsing for Sentiment analysis", Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 1533-1541, Singapore, 6-7 August 2009