

# Sentiment Analysis on Hadoop with Hadoop Streaming

Piyush Gupta  
Research Scholar  
Dept. of Comp. Sci, and Appl.  
KUK, Haryana

Pardeep Kumar  
Assistant Professor  
Dept. of Comp. Sci, and Appl.  
KUK, Haryana

Girdhar Gopal  
Assistant Professor  
Dept. of Comp. Sci, and Appl.  
KUK, Haryana

## ABSTRACT

Ideas and opinions of peoples are influenced by the opinions of other peoples. Lot of research is going on analysis of reviews given by peoples. Sentiment analysis is the major computational technique to calculate or observe sentiments of people's thoughts. Therefore, a method that assigns scores indicating positive and negative opinion about the product is proposed. It uses Hadoop Distributed File System (HDFS) to store data set and run on MapReduce architecture for performing sentiment analysis.

## General Terms

Sentiment Analysis, hadoop streaming.

## Keywords

Hadoop, MapReduce, Opinion, Sentiment Analysis.

## 1. INTRODUCTION

Some well-known internet companies like Google, Amazon, LinkedIn, Yahoo! etc have generated a huge amount of structured and unstructured data every day. This exponential growth of data leads to some challenges like processing of large data sets, extraction of useful information from online generated data sets etc. Hadoop is one of the processing tools that is used to analyse and process large data sets. It has two main components: HDFS, Hadoop Distributed File System used for reliable storage of data and MapReduce, which is used to process the data. Hadoop's MapReduce and HDFS components were inspired by Google technologies on MapReduce and GoogleFileSystem.

Hadoop framework is used for sentiment analysis in this paper. MapReduce program for hadoop are written in python language for doing sentiment analysis. Even though hadoop framework is written in java, programs for hadoop need not to be coded only in java but can also be developed in other language like python or c++ using the hadoop streaming interface. With hadoop streaming, programs that acts as the mapper and a program that acts as a reducer needs to be written. These applications must interface with input-output streams in such a way equivalent to the following series of pipes in shell environment [1]:

```
$ cat input_file.txt | ./mapper_name.py | sort |  
./reducer_name.py > output_file.txt
```

Through hadoop streaming, mapper will read data from STDIN then sends the mapped key/value pairs to the reducer via STDIN and then output its result to STDOUT.

This paper is organized as follows: section 2 describes the related work that has been done in past for sentiment analysis; proposed approach for this research is explained in section 3, which is followed by implementations and results in section 4;

finally the conclusions and findings of this research are explained in section 5.

## 2. RELATED WORK

In the past years, many works has been released in sentiment analysis. For sentiment analysis, there exist many possible variants; some of them are discussed in following section.

Batool, R. et al. [2] analyzed tweets to classify data and sentiments from Twitter more precisely. The information from tweets is extracted using keyword based knowledge extraction. Moreover, the extracted knowledge is further enhanced using domain specific seed based enrichment technique. The proposed methodology facilitates the extraction of keywords, entities, synonyms, and parts of speech from tweets which are then used for tweets classification and sentiment analysis. The proposed system is tested on a collection of 40,000 tweets. By applying the Knowledge Enhancer and Synonym Binder module on the extracted information the authors have achieved increase in information gain in a range of 0.1% to 55%.

Xiaoqian Zhang et al. [3] illustrated that a corpus which consists of polarity-shifted sentences in various kinds of product reviews is created. In the corpus, both the sentimental words and shifting trigger words are annotated. Furthermore, all the polarity shifted sentences are analyzed and categorize them into five categories: opinion-itself, holder target, time and hypothesis. Experimental study shows the agreement of annotation and the distribution of the five categories of polarity shifting.

Luo Yi et al. [4] introduced a novel approach named Feature-Grading which is a comprehensive algorithm used to make recommendation of commodities in e-commerce business. This technique is based on the integration of feature mining, sentiment analysis, and the records of customer historical behaviors. The process of Feature-Grading is separated into five key steps: (a) Extracting overall feature set of a group category of commodities; (b)Extracting modifier set and negative words set; (c)Acquiring specific feature set and feature assessment set; (d)Acquiring specific feature weight set; (e)Acquiring item weight set. Then rank and grade all the items with an acquired grading equation and needed as well as top ranking items can be recommended. The authors utilize the real information of mobiles and their reviews from the famous e-commerce website Amazon.cn as an experimental data.

Kumar Singh, P. et al. [5] aimed to automate the process of gathering online, end user reviews for any given product or service and analyzing those reviews in terms of the sentiments expressed about specific features. This involves the filtering of irrelevant and unhelpful reviews, quantification of the sentiments of thousands of (useful) reviews. And finally, providing the end user (business/manufacturer) summarized

data about the expressed sentiments in way of intuitive and easy to understand graphs, charts and other visualization. This data can then be used to improve business outcomes and ensure a very high level of customer satisfaction.

Hui Song et al. [6] proposed an approach based on patterns for extracting product features from online reviews for sentiment analysis. To enhance adaptability of the pattern set, authors merge some fundamental patterns into a new fuzzy pattern. Then a pattern matching algorithm is applied to extract the titles and opinion words from the reviews. The authors conducted a platform to extract features from product reviews automatically.

Bin Wen et al. [7] aimed to obtain the useful orientation information especially in Micro-blog and mainly research focus in the nature language processing. The authors presents a sentence orientation identification method taking advantage of an improved algorithm for calculating Chinese term semantic orientation computation, then, combining adverbs and conjunctions points a sentence identification algorithm.

Shoushan Li et al. [8] aimed to perform sentiment classification with full consideration of the polarity shifting phenomenon. The authors first extract some detection rules for detecting polarity shifting of sentimental words from a corpus which consists of polarity-shifted sentences. Then using these detection rules they detect the polarity-shifted words in the testing data. Third, a novel term counting-based classifier is designed by fully considering those polarity-shifted words. The authors show that the novel term counting-based classifier significantly improves the performance of sentiment analysis across five domains.

Jamoussi, S and Ameer, H. [9] determined the sentiment orientation of a Facebook comment (positive or negative) by using the linguistic approach. As sentiment lexicon plays a key role in sentiment analysis applications so authors create a lexicon covering several sentiment words. The authors addresses the problem how to group and list words present in the corpus into two dictionaries that exploits the emotions symbols( emoticons, acronyms and exclamation words) present in comments. Finally, by using these prepared dictionaries, they predict the positive and negative polarities of the comment.

### 3. PROPOSED APPROACH

In this section, a method to calculate sentiment score of reviews given by the customers is proposed and implemented in python on hadoop. The method works in two phases: In first phase, Mapper will parse the given input file and in second phase, Reducer will calculate the sentiments. The data set we used contains different types of clothing product reviews from Amazon.com [10]. We present some statistics about the data set in Table I. For instance, number of reviews about the product, number of positive and negative words, number of stop words. We use a positive and negative word dictionary to identify positive and negative words [11] [12]. Stop word dictionary is used to identify and remove stop words from the reviewed product [13].

Table 1: Statistics about the data set

Number of Reviews	1680
Number of stop words	538
Number of positive words	4784
Number of negative words	2005

The architecture of proposed system is shown in Figure 1. Hadoop is one of the processing tools that is used to analyse and process large data sets. It has two main components: MapReduce, which is used to process the data and HDFS, Hadoop Distributed File System used for reliable storage of data. Hadoop's MapReduce and HDFS components were inspired by Google papers on their MapReduce and GoogleFileSystem. The input and output data file is stored in HDFS. Mapper and reducer are used to parse the file and to compute sentiments respectively. The input data file consists of key/value pair records. The details of each component are described in the following subsections:

#### 3.1 HDFS (Hadoop Distributed File System)

Hadoop uses HDFS to store files efficiently in the cluster. When a file is placed in HDFS it is broken down into blocks, minimum of 64 MB block size which is far more in comparison to other file systems like FAT, NTFS where the block size is typically 4-32 KB. Each block is assigned a number blk\_XXXXXX, where XXXXXX is a long number depending on the size of data. Each of these blocks is then saved on some nodes in the cluster called Data Nodes. Meta Data of these files is stored on a specific node called Name Node in the cluster. To prevent the failure at data nodes, these blocks are replicated across different nodes in the cluster. The default replication value is 3. A Hadoop cluster can comprise of a single node (single node cluster) or thousands or millions of nodes.

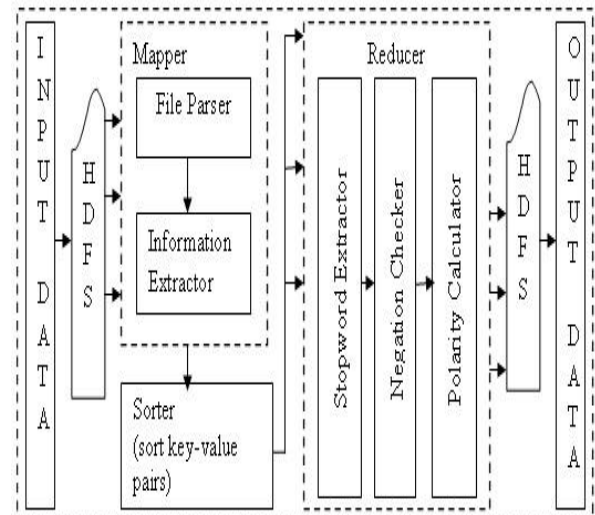


Fig 1: Proposed architecture for sentiment analysis

#### 3.2 Mapper

It maps input key/value pairs to a set of intermediate key/value pairs [14]. Mapper will prepare the data for processing in the Reducer.

**File parser:** This will split the data into records and records in turn will split into individual key-value pair.

**Information Extractor:** It is used to get the key/value pairs that are necessary to do the sentiment analysis.

#### 3.3 Sorter

It will sort the intermediate key/value pairs generated by the mapper. This sorted key/value pairs acts as the input to the Reducer phase.

### 3.4 Reducers

It will do the sentiment analysis of coming input file. It works in 3 steps:

**Stop word extractor:** Stop words like which, after, has, become are those words which are not participate in the sentiment analysis, so removal of these words will not affect the experimental results.

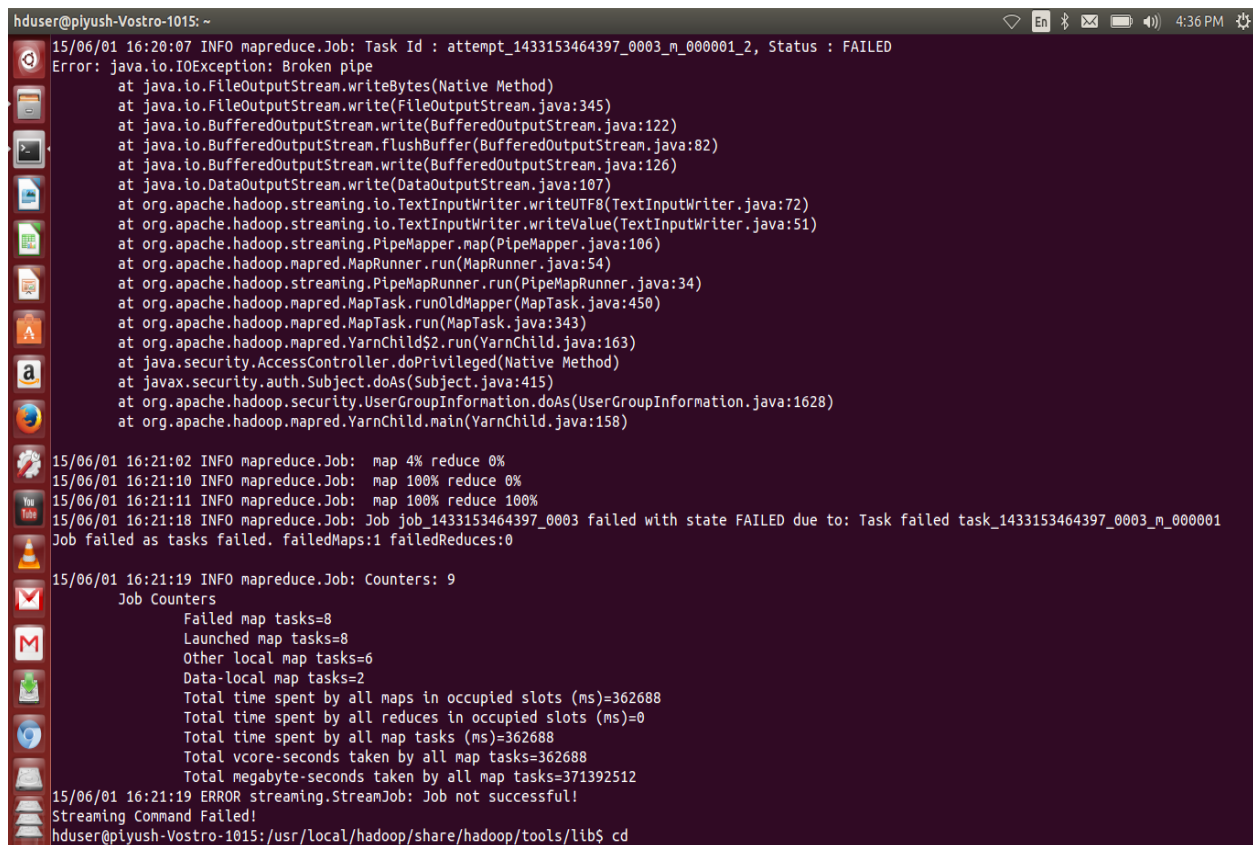
**Negation checker:** Usually negation will change the polarity of positive and negative words. For instance, “this product is not good” – in this sentence good is a positive word but if it is used with not it becomes negative [15]. So we reverse the polarity of a word whenever it is preceded by a negation. We increase/decrease the polarity strength when a word is preceded by a negation. Thus not good = -1; good = 1; not bad = +1; bad = -1.

**Polarity calculator:** The polarity calculator is basically the main part of proposed framework. It uses positive, negative

and stop-word dictionary to extract the sentiment-oriented words from each sentence. The Polarity Calculator identifies whether a sentence is positive or negative. It will calculate the positive polarity and negative polarity of a sentence, then subtracting these polarities, if we get negative number then overall sentence is negative otherwise positive. After calculating polarity the output will be stored in HDFS.

## 4. IMPLEMENTATION AND RESULT

The proposed approach is implemented on Amazon product reviews dataset [10]. Total number of records and file size is explained in table 1 above. The hadoop is maintained both as a single node cluster as well as multi node cluster machines. The single node cluster machine fails to give the output as the data size is out of bounds for its physical memory to process, so setup of multi node cluster is created and the work is done on that cluster. Figure 2 and figure 3 shows single cluster and multi cluster results respectively.



```
hduser@piyush-Vostro-1015: ~
15/06/01 16:20:07 INFO mapreduce.Job: Task Id : attempt_1433153464397_0003_m_000001_2, Status : FAILED
Error: java.io.IOException: Broken pipe
    at java.io.FileOutputStream.writeBytes(Native Method)
    at java.io.FileOutputStream.write(FileOutputStream.java:345)
    at java.io.BufferedOutputStream.write(BufferedOutputStream.java:122)
    at java.io.BufferedOutputStream.flushBuffer(BufferedOutputStream.java:82)
    at java.io.BufferedOutputStream.write(BufferedOutputStream.java:126)
    at java.io.DataOutputStream.write(DataOutputStream.java:107)
    at org.apache.hadoop.streaming.io.TextInputWriter.writeUTF8(TextInputWriter.java:72)
    at org.apache.hadoop.streaming.io.TextInputWriter.writeValue(TextInputWriter.java:51)
    at org.apache.hadoop.streaming.PipeMapper.map(PipeMapper.java:106)
    at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:54)
    at org.apache.hadoop.streaming.PipeMapRunner.run(PipeMapRunner.java:34)
    at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:450)
    at org.apache.hadoop.mapred.MapTask.run(MapTask.java:343)
    at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:163)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:415)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1628)
    at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:158)

15/06/01 16:21:02 INFO mapreduce.Job: map 4% reduce 0%
15/06/01 16:21:10 INFO mapreduce.Job: map 100% reduce 0%
15/06/01 16:21:11 INFO mapreduce.Job: map 100% reduce 100%
15/06/01 16:21:18 INFO mapreduce.Job: Job job_1433153464397_0003 failed with state FAILED due to: Task failed task_1433153464397_0003_m_000001
Job failed as tasks failed. failedMaps:1 failedReduces:0

15/06/01 16:21:19 INFO mapreduce.Job: Counters: 9
Job Counters
    Failed map tasks=8
    Launched map tasks=8
    Other local map tasks=6
    Data-local map tasks=2
    Total time spent by all maps in occupied slots (ms)=362688
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=362688
    Total vcore-seconds taken by all map tasks=362688
    Total megabyte-seconds taken by all map tasks=371392512
15/06/01 16:21:19 ERROR streaming.StreamJob: Job not successful!
Streaming Command Failed!
hduser@piyush-Vostro-1015: /usr/local/hadoop/share/hadoop/tools/lib$ cd
```

Fig 2: Job halted unsuccessfully on single cluster machine

```

hduser@piyush-Vostro-1015: /usr/local/hadoop/share/hadoop/tools/lib
15/06/01 14:40:19 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
packageJobJar: [/tmp/hadoop-unjar8362184257589583570/] [] /tmp/streamjob12429048
64488553247.jar tmpDir=null
15/06/01 14:40:21 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0
:8032
15/06/01 14:40:22 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0
:8032
15/06/01 14:40:25 INFO mapred.FileInputFormat: Total input paths to process : 1
15/06/01 14:40:26 INFO mapreduce.JobSubmitter: number of splits:2
15/06/01 14:40:27 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_14
33148764393_0002
15/06/01 14:40:28 INFO impl.YarnClientImpl: Submitted application application_1433148764393_0002
15/06/01 14:40:29 INFO mapreduce.Job: The url to track the job: http://piyush-Vostro-1015:8088/proxy/application_1433148764393_0002/
15/06/01 14:40:29 INFO mapreduce.Job: Running job: job_1433148764393_0002
15/06/01 14:41:01 INFO mapreduce.Job: Job job_1433148764393_0002 running in uber mode : false
15/06/01 14:41:01 INFO mapreduce.Job: map 0% reduce 0%
15/06/01 14:42:58 INFO mapreduce.Job: map 33% reduce 0%
15/06/01 14:43:02 INFO mapreduce.Job: map 67% reduce 0%
15/06/01 14:43:05 INFO mapreduce.Job: map 100% reduce 0%
15/06/01 14:47:19 INFO mapreduce.Job: map 100% reduce 100%
15/06/01 14:47:25 INFO mapreduce.Job: Job job_1433148764393_0002 completed successfully
15/06/01 14:47:29 INFO mapreduce.Job: Counters: 49
File System Counters
FILE: Number of bytes read=21422
FILE: Number of bytes written=366270
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=20913
HDFS: Number of bytes written=345
HDFS: Number of read operations=9
HDFS: Number of large read operations=0
HDFS: Number of write operations=2
Job Counters
Launched map tasks=2
Launched reduce tasks=1
Data-local map tasks=2
Total time spent by all maps in occupied slots (ms)=276973
Total time spent by all reduces in occupied slots (ms)=211019
Total time spent by all map tasks (ms)=276973

```

Fig 3: Job completed successfully on multi cluster machine

Table 2: Analytics done by the proposed algorithm

Reviews	Proposed system	Manual system	Error rate
% of positive reviews	88.03	82.14	5.89
% of Negative reviews	31.13	28.75	2.38

**Accuracy Test.** A test of the sentiment analysis has been conducted to measure accuracy. Accuracy is performed by using the review scores of a product given by customers. First total number of positive and negative reviews is calculated by manually examining score of a review i.e. if score is greater than or equal to 3 then it is positive review otherwise it is a negative review. Then the percentage of positive and negative reviews is calculated for both proposed and manual system as shown in table 2 and figure 4. The error rate for positive and negative reviews are relatively small.

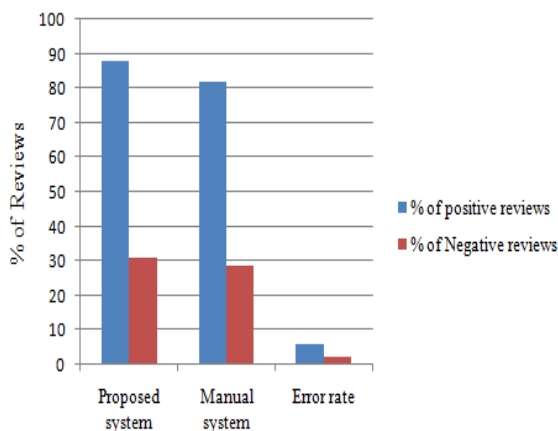


Fig 4: Analytics result with proposed algorithm

## 5. CONCLUSION

As the data is growing rapidly nowadays, it is required to process this data at a fairly fast speed. For which hadoop and MapReduce are two hot and current technologies to be used. Sentiment Analysis is one of the analytics which will tell the writers/producers sentiment about their thoughts. What they think about the particular entity and what they have described? It is a routine task in these days to find out the peoples sentiment about an entity of real world. A new approach about doing the sentiment analysis is proposed in this paper with the use of MapReduce to run it faster. The analytics result shows that it does a great job in finding the sentiment and it runs in parallel to give the results faster.

## 6. REFERENCES

- [1] "Writing Hadoop Applications in Python," (Last visited in June 2015). [online]. Available: <http://www.glennklockwood.com/di/hadoop-streaming.php>
- [2] Batool, R.; Khattak, A.M.; Maqbool, J.; Sungyoung Lee, "Precise tweet classification and sentiment analysis," *Computer and Information Science (ICIS), 2013 IEEE/ACIS 12th International Conference on*, vol., no., pp.461,466, 16-20 June 2013
- [3] Xiaoqian Zhang; Shoushan Li; Guodong Zhou; Hongxia Zhao, "Polarity Shifting: Corpus Construction and Analysis," *Asian Language Processing (IALP), 2011 International Conference on*, vol., no., pp.272,275, 15-17 Nov. 2011

- [4] Luo Yi; Fan Miao; Zhou Xiaoxia, "The design and implementation of Feature-Grading recommendation system for e-commerce," *Information and Automation (ICIA), 2011 IEEE International Conference on* , vol., no., pp.236,241, 6-8 June 2011
- [5] Kumar Singh, P.; Sachdeva, A.; Mahajan, D.; Pande, N.; Sharma, A., "An approach towards feature specific opinion mining and sentimental analysis across e-commerce websites," *Confluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference -* , vol., no., pp.329,335, 25-26 Sept. 2014
- [6] Hui Song; Yingxiang Fan; Xiaoqiang Liu; Dao Tao, "Extracting product features from online reviews for sentimental analysis," *Computer Sciences and Convergence Information Technology (ICCIT), 2011 6th International Conference on* , vol., no., pp.745,750, Nov. 29 2011-Dec. 1 2011
- [7] Bin Wen; Wenhua Dai; Junzhe Zhao, "Sentence Sentimental Classification Based on Semantic Comprehension," *Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on* , vol.2, no., pp.458,461, 28-29 Oct. 2012
- [8] Shoushan Li; Zhongqing Wang; Lee, S.Y.M.; Chu-Ren Huang, "Sentiment Classification with Polarity Shifting Detection," *Asian Language Processing (IALP), 2013 International Conference on* , vol., no., pp.129,132, 17-19 Aug. 2013
- [9] Jamoussi, S.; Ameer, H., "Dynamic construction of dictionaries for sentiment classification," *Cloud and Green Computing (CGC), 2013 Third International Conference on* , vol., no., pp.418,425, Sept. 30 2013-Oct. 2 2013
- [10] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. *RecSys*, 2013.
- [11] "Opinion Mining, Sentiment Analysis, and Opinion Spam Detection," (Last visited in June 2015) [online]. Available: <http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>
- [12] Minqing Hu and Bing Liu. "Mining and Summarizing Customer Reviews." *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, Aug 22-25, 2004, Seattle, Washington, USA.
- [13] "stop-words," (Last visited in June 2015) [online]. Available: <https://code.google.com/p/stop-words/>
- [14] "Interface Mapper," (Last visited in June 2015) [online]. Available: <http://hadoop.apache.org/docs/r2.6.0/api/org/apache/hadoop/mapred/Mapper.html>
- [15] Godbole, N., Srinivasaiah, M., & Skiena, S. (2007). "Large-Scale Sentiment Analysis for News and Blogs." *ICWSM 7 (2007)*: 21
- [16] Asmi, A., & Ishaya, T. "Negation identification and calculation in sentiment analysis." *IMMM 2012, The Second International Conference on Advances in Information Mining and Management*. 2012