Journal of Theoretical and Applied Information Technology
15th June 2018. Vol.96. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

THE MULTI-DIMENSIONAL VECTORS AND AN YULE-II MEASURE USED FOR A SELF-ORGANIZING MAP ALGORITHM OF ENGLISH SENTIMENT CLASSIFICATION IN A DISTRIBUTED ENVIRONMENT

[1]DR.VO NGOC PHU, [2]DR.VO THI NGOC TRAN

[1]Nguyen Tat Thanh University, 300A Nguyen Tat Thanh Street, Ward 13, District 4, Ho Chi Minh City, 702000, Vietnam

[2]School of Industrial Management (SIM), Ho Chi Minh City University of Technology - HCMUT, Vietnam National University, Ho Chi Minh City, Vietnam

E-mail: [1]vongocphu03hca@gmail.com, vongocphu@ntt.edu.vn, [2]vtntran@HCMUT.edu.vn

ABSTRACT

We have proposed a new model for big data sentiment classification using a Self-Organizing Map Algorithm (SOM) – an unsupervised learning of a machine learning to classify the sentiments (positive, negative, or neutral) for all the documents of our testing data set according to all the documents of our training data set in English. We only run the SOM only once, the results of the sentiment classification of all the documents of the testing data are identified. The SOM is proposed according to many multi-dimensional vectors of both the testing data set and the training data set. The multi-dimensional vectors are based on many sentiment lexicons of our basis English sentiment dictionary (bESD). One document is corresponding to one multi-dimensional vector according to the sentiment lexicons. After running the SOM only once, a Map is used in presenting the results of the SOM. The results of clustering all documents of the testing data set into either the positive polarity or the negative polarity are shown on the Map, we can find all the results of the sentiment classification of all the documents of the testing data set fully. We only use many multi-dimensional vectors based on the sentiment lexicons of the bESD. In a sequential system, the new model has been tested firstly, and then, this model has been performed in a parallel network environment secondly. The accuracy of the testing data set has been achieved 88.72% certainly. Many different fields can widely use the results of this new model.

Keywords: *English sentiment classification; parallel system; Cloudera; Hadoop Map and Hadoop Reduce; Yule-II Measure; Self-Organizing Map*

1. INTRODUCTION

Sentiment classification has already been studied for many years. Many significant contributions of the sentiment classification have already been found certainly.

A clustering data is a set of objects which is processed into classes of similar objects in a data mining field. One cluster is a set of data objects which are similar to each other and are not similar to objects in other clusters. A number of data clusters can be clustered, which can be identified following experience or can be automatically identified as part of clustering method. There are the surveys in [1-14] related to both the data mining and the natural language processing certainly.

A Self-Organizing Map Algorithm (SOM) based on the survey in [15-19] provides the information as follows: Self-organizing neural networks are used to cluster input patterns into groups of similar patterns. They're called "maps" because they assume a topological structure among their cluster units; effectively mapping weights to input data. The Kohonen network is probably the best example, because it's simple, yet introduces the concepts of self-organization and unsupervised learning easily. The Self-Organizing Map (SOM), commonly also known as Kohonen network (Kohonen 1982, Kohonen 2001) is a computational method for the visualization and analysis of high-dimensional data, especially experimentally acquired information. The advantages of the SOM are as follows: It is an unsupervised learning. We do not need any training data sets in English for the SOM. It shows many multi-dimensional data sets

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

into either the one-dimensional data sets or the two-dimensional data sets, and etc.

The basic principles are proposed for our new model as follows:

1)It is assumed that each English sentence has m English words (or English phrases).

2)It is assumed that the maximum number of one English sentence is m_max terms (words or phrases); it means that m is less than m_max or m is equal to m_max.

3)It is assumed that each English document has n English sentences.

4)Assuming that the maximum number of one English document is n_max sentences; it means that n is less than n_max or n is equal to n_max.

The motivation of this new model is as follows: Many algorithms in the data mining field can be applied to natural language processing, specifically semantic classification for processing millions of English documents. The SOM and an Yule-II Measure (YIIM) of the clustering technologies of the data mining filed can be applied to the sentiment classification in both a sequential environment and a parallel network system. This will result in many discoveries in scientific research, hence the motivation for this study.

The novelty of the proposed approach is as follows: the SOM and the YIIM are applied to sentiment analysis. This can also be applied to identify the sentiments (positive, negative, or neutral) of millions of many documents. This survey can be applied to other parallel network systems. Hadoop Map (M) and Hadoop Reduce (R) are used in the proposed model. Therefore, we will study this model in more detail.

According to the purpose of the research, we always try to find a new approach to improve many accuracies of the results of the sentiment classification and to shorten many execution times of the proposed model with a low cost.

To get higher accuracy and shorten execution time of the sentiment classification, we do not use a vector space modeling (VSM) in [1-3]. We use many sentiment lexicons of our basis English sentiment dictionary (bESD). We do not use any one-dimensional vectors based on both the VSM [1-3] and the sentiment lexicons. We also do not use any multi-dimensional vectors according to the VSM [1-3]. We only use many multi-dimensional vectors based on the sentiment lexicons of the bESD. The sentiment lexicons of the bESD are identified by using the YIIM through a Google search engine with AND operator and OR operator. We transfer one document into one multi-dimensional vector according to the sentiment lexicons of the bESD. All the documents of the testing data set are transferred into the multi-dimensional vectors based on the sentiment lexicons. The positive multi-dimensional vectors which we transfer all the positive documents of the training data set into the positive multi-dimensional vectors. A positive multi-dimensional central vector is a central vector of the multi-dimensional vectors of the positive multi-dimensional vectors of the training data set. The negative multi-dimensional vectors which we transfer all the negative documents of the training data set into the negative multi-dimensional vectors. A negative multi-dimensional central vector is a central vector of the multi-dimensional vectors of the negative multi-dimensional vectors of the training data set. The SOM is proposed according to many multi-dimensional vectors of both the testing data set and the training data set. We only run the SOM only once, the results of the sentiment classification of all the documents of the testing data are identified completely. After running the SOM only once, a Map is used in presenting the results of the SOM. The results of clustering all the documents of the testing data set into either the positive polarity or the negative polarity are shown on the Map, we can find all the results of the sentiment classification of all the documents of the testing data set fully.

In this survey, we implement the proposed model as follows: Firstly, we create the sentiment lexicons of the bESD which the valences and the polarities of them are calculated by using the YIIM through the Google search engine with AND operator and OR operator. All the documents of the testing data set are transferred into the multi-dimensional vectors of the testing data set based on the sentiment lexicons of the bESD. The positive documents of the training data set are transferred into the multi-dimensional vectors based on the sentiment lexicons of the bESD, called the positive vector group of the training data set. Then, we identify one positive multi-dimensional central vector which is a central vector of the positive vector group. The negative documents of the training data set are transferred into the multi-dimensional vectors based on the sentiment lexicons of the bESD, called the negative vector group of the training data set. We calculate one negative multi-dimensional central vector which is a central vector of the negative vector group. All the multi-mensional vectors of the testing data set are clustered into either the positive polarity or the negative polarity by using the SOM with the input is all the multi-dimensional vectors of the testing data set. The SOM uses a Map which is a matrix with its rows are the number of the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

documents of the testing data set and its columns is 2. We set an initialization of the SOM with its map in Figure 1 as follows:

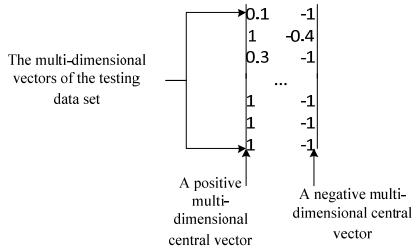

Figure 1: An initialization of the SOM – the Map

Then, after the SOM is implemented completely, we have the Map in Figure 2 as follows:

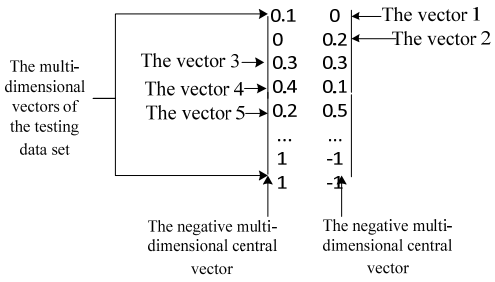

Figure 2: The final Map – the result of clustering by using the SOM

In Figure 2, we have the multi-dimensional vector 1 (0.1, 0), the multi-dimensional vector 2 (0, 0.2), the multi-dimensional vector 3 (0.3, 0.3), the multi-dimensional vector 4 (0.4, 0.1) , and the multi-dimensional vector 5 (0.2, 0.5) on the Map as follows: With the multi-dimensional vector 1 (corresponding to the document 1 of the testing data set), the column of the positive polarity is 0.1 and the column of the negative polarity is 0, Thus, the value of the column of the positive polarity is greater than the value of the column of the negative polarity, therefore this multi-dimensional vector is clustered into the positive. With the multi-dimensional vector 2 (corresponding to the document 2 of the testing data set), the column of the positive polarity is 0 and the column of the negative polarity is 0.2. Therefore, the value of the column of the positive polarity is less than the value of the column of the negative polarity, thus this multi-dimensional vector is clustered into the negative. With the multi-dimensional vector 3 (corresponding to the document 3 of the testing data set), the column of the positive polarity is 0.3 and the column of the negative polarity is 0.3. So, the value of the column of the positive polarity is as equal as the value of the column of the negative polarity, therefore this multi-dimensional vector is not clustered into both the positive and the negative. It means that this multi-dimensional vector is clustered into the neutral polarity. With the multi-dimensional vector 4 (0.4, 0.1) (corresponding to the document 4 of the testing data set), the column of the positive polarity is 0.4 and the column of the negative polarity is 0.1. Thus, the value of the column of the positive polarity is greater than the value of the column of the negative polarity, so this multi-dimensional vector is clustered into the positive. With the multi-dimensional vector 5 (corresponding to the document 5 of the testing data set), the column of the positive polarity is 0.2 and the column of the negative polarity is 0.5. Therefore, the value of the column of the positive polarity is less than the column of the negative polarity, thus this multi-dimensional vector is clustered into the negative. Finally, the sentiment classification of all the documents of the testing data set is identified completely.

All the above things are firstly implemented in the sequential system to get an accuracy of the result of the sentiment classification and an execution time of the result of the sentiment classification of the proposed model. All the above things are secondly performed in the parallel network environment to shorten the execution times of the proposed model to get the accuracy of the results of the sentiment classification and the execution times of the results of the sentiment classification of our new model

The significant contributions of our new model can be applied to many areas of research as well as commercial applications as follows:

1)Many surveys and commercial applications can use the results of this work in a significant way.

3)The algorithms are built in the proposed model.

4)This survey can certainly be applied to other languages easily.

5)The results of this study can significantly be applied to the types of other words in English.

6)Many crucial contributions are listed in the Future Work section.

7)The algorithm of data mining is applicable to semantic analysis of natural language processing.

8)This study also proves that different fields of scientific research can be related in many ways.

9)Millions of English documents are successfully processed for emotional analysis.

10)The semantic classification is implemented in the parallel network environment.

11)The principles are proposed in the research.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

12)The Cloudera distributed environment is used in this study.
13)The proposed work can be applied to other distributed systems.
14)This survey uses Hadoop Map (M) and Hadoop Reduce (R).
15)Our proposed model can be applied to many different parallel network environments such as a Cloudera system
16)This study can be applied to many different distributed functions such as Hadoop Map (M) and Hadoop Reduce (R).
17)The SOM – related algorithms are proposed in this survey.
18)The YIIM – related algorithms are built in this work.
This study contains 6 sections. Section 1 introduces the study; Section 2 discusses the related works about the vector space modeling (VSM), Self-Organizing Map Algorithm (SOM), Yule-II Measure (YIIM), etc.; Section 3 is about the English data set; Section 4 represents the methodology of our proposed model; Section 5 represents the experiment. Section 6 provides the conclusion. The References section comprises all the reference documents; all tables are shown in the Appendices section.

2. RELATED WORK

We summarize many researches which are related to our research.

There are the works related to vector space modeling (VSM) in [1-3]. In this study [1], the authors examined the Vector Space Model, an Information Retrieval technique and its variation. In this survey [2], the authors consider multi-label text classification task and apply various feature sets. The authors consider a subset of multi-labeled files from the Reuters-21578 corpus. The authors use traditional tf-IDF values of the features and tried both considering and ignoring stop words. The authors also tried several combinations of features, like bigrams and unigrams. The authors in [3] introduce a new weighting method based on statistical estimation of the importance of a word for a specific categorization problem. This method also has the benefit to make feature selection implicit, since useless features for the categorization problem considered get a very small weight.

The latest researches of the sentiment classification are [4-14]. In the research [4], the authors present their machine learning experiments with regard to sentiment analysis in blog, review and forum texts found on the World Wide Web and written in English, Dutch and French. The survey in [5] discusses an approach where an exposed stream of tweets from the Twitter micro blogging site are preprocessed and classified based on their sentiments. In sentiment classification system the concept of opinion subjectivity has been accounted. In the study, the authors present opinion detection and organization subsystem, which have already been integrated into our larger question-answering system, etc.

There are the researches related the Self-Organizing Map Algorithm (SOM) in [15-19]. In [15], the self-organized map, an architecture suggested for artificial neural networks, is explained by presenting simulation experiments and practical applications. The self-organizing map has the property of effectively creating spatially organized internal representations of various features of input signals and their abstractions. In [16], the Kohonen Self-Organizing Map (SOM) is one of the most well-known neural network with unsupervised learning rules; it performs a topology-preserving projection of the data space onto a regular two-dimensional space. Its achievement has already been demonstrated in various areas, but this approach is not yet widely known and used by ecologists. The present work describes how SOM can be used for the study of ecological communities, etc.

By far, we know that PMI (Pointwise Mutual Information) equation and SO (Sentiment Orientation) equation are used for determining polarity of one word (or one phrase), and strength of sentiment orientation of this word (or this phrase). Jaccard measure (JM) is also used for calculating polarity of one word and the equations from this Jaccard measure are also used for calculating strength of sentiment orientation this word in other research. PMI, Jaccard, Cosine, Ochiai, Tanimoto, and Sorensen measure are the similarity measure between two words; from those, we prove that the Yule-II (YIIM) is also used for identifying valence and polarity of one English word (or one English phrase). Finally, we identify the sentimental values of English verb phrases based on the basis English semantic lexicons of the basis English emotional dictionary (bESD).

There are the works related to PMI measure in [20-32]. In the research [20], the authors generate several Norwegian sentiment lexicons by extracting sentiment information from two different types of Norwegian text corpus, namely, news corpus and

discussion forums. The methodology is based on the Point wise Mutual Information (PMI). The authors introduce a modification of the PMI that considers small "blocks" of the text instead of the text as a whole. The study in [21] introduces a simple algorithm for unsupervised learning of semantic orientation from extremely large corpora, etc.

Two studies related to the PMI measure and Jaccard measure are in [33, 34]. In the survey [33], the authors empirically evaluate the performance of different corpora in sentiment similarity measurement, which is the fundamental task for word polarity classification. The research in [34] proposes a new method to estimate impression of short sentences considering adjectives. In the proposed system, first, an input sentence is analyzed and preprocessed to obtain keywords. Next, adjectives are taken out from the data which is queried from Google N-gram corpus using keywords-based templates.

The works related to the Jaccard measure are in [35-41]. The survey in [35] investigates the problem of sentiment analysis of the online review. In the study [36], the authors are addressing the issue of spreading public concern about epidemics. Public concern about a communicable disease can be seen as a problem of its own, etc.

The surveys related the similarity coefficients to calculate the valences of words are in [47-51].

The English dictionaries are [52-57] and there are more than 55,000 English words (including English nouns, English adjectives, English verbs, etc.) from them.

There are the works related to the Yule-II (YIIM) in [58-63]. The authors in [58] collected 76 binary similarity and distance measures used over the last century and reveal their correlations through the hierarchical clustering technique. The Yule coefficient in [59] is used to measure the skewness of a frequency distribution, etc.

3. DATA SET

Based on Figure 3 below, we built our the testing data set including the 5,500,000 documents in the movie field, which contains the 2,750,000 positive and 2,750,000 negative in English. All the documents in our English testing data set are automatically extracted from English Facebook, English websites and social networks; then we labeled positive and negative for them.

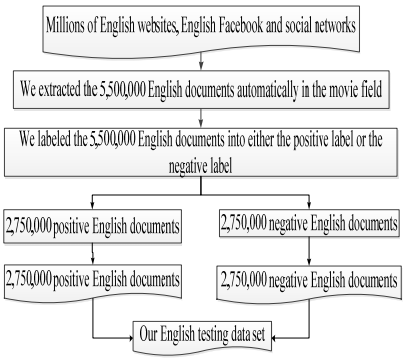

Figure 3: Our testing data set in English.

In Figure 4 below, we built our the training data set including the 3,000,000 documents in the movie field, which contains the 1,500,000 positive and 1,500,000 negative in English. All the documents in our English training data set are automatically extracted from English Facebook, English websites and social networks; then we labeled positive and negative for them.

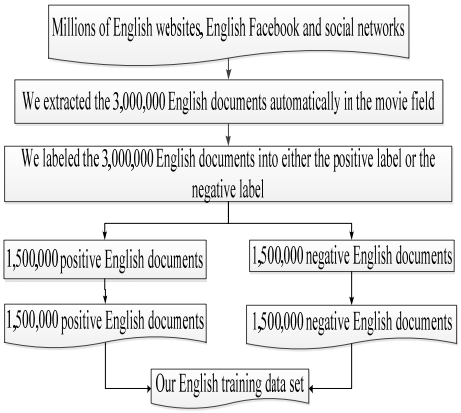

Figure 4: Our training data set in English.

4. METHODOLOGY

This section comprises two sub-sections as follows: (4.1) and (4.2). In the sub-section (4.1), we use the SOM to cluster the documents of the testing data set into either the positive or the negative in both a sequential environment and a parallel distributed system. In the sub-section (4.2), we create the sentiment lexicons of the bESD.

The sub-section (4.1) includes two sub-sections. The first sub-section is the sub-section (4.1.1) which the SOM is used in clustering the documents of the testing data set into either the positive or the negative in a sequential environment. The second sub-section is the sub-section (4.1.2) which the SOM is used in clustering the documents of the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

testing data set into either the positive or the negative in a parallel network system.

The sub-section (4.2) has three parts as follows: (4.2.1), (4.2.2) and (4.2.3). In the first part (4.2.1), we identify a sentiment value of one word (or one phrase) in English. In the second part (4.2.2), we create a basis English sentiment dictionary (bESD) in a sequential system. In the third part (4.2.3), we create a basis English sentiment dictionary (bESD) in a parallel environment.

4.1 Using the SOM to cluster the documents of the testing data set into either the positive or the negative in both a sequential environment and a parallel distributed system

In Figure 5, an overview of the proposed model is presented. There are two sub-section of this section. The first sub-section is the sub-section (4.1.1) which the SOM is used in clustering the documents of the testing data set into either the positive or the negative in a sequential environment. The second sub-section is the sub-section (4.1.2) which the SOM is used in clustering the documents of the testing data set into either the positive or the negative in a parallel network system.

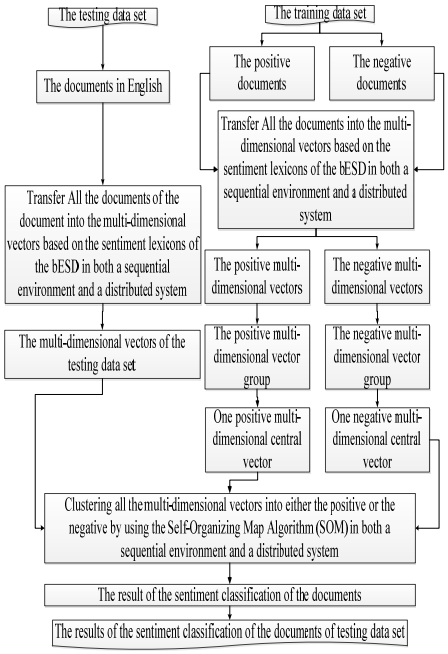

Figure 5: Overview of our new model.

4.1.1 Using Self-Organizing Map Algorithm to cluster the documents of the testing data set into either the positive or the negative in a sequential environment

The Self-Organizing Map Algorithm is used in clustering the documents of the testing data set into either the positive or the negative in a sequential environment in Figure 6 as follows:

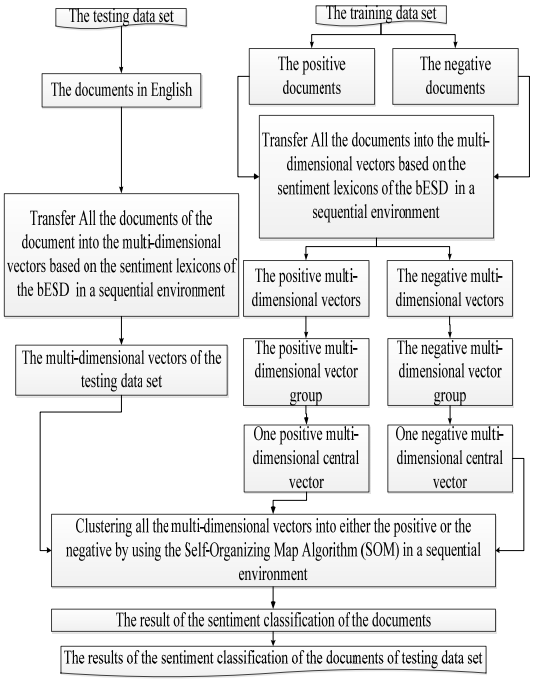

Figure 6: Overview of using Self-Organizing Map Algorithm to cluster the documents of the testing data set into either the positive or the negative in a sequential environment

We perform this section in a sequential environment in Figure 6 as follows: According to the creating a basis English sentiment dictionary (bESD) in a sequential environment in (4.2.2), we calculate the sentiment values and polarities of the sentiment lexicons. All the documents of the testing data set are transferred into the multi-dimensional vectors of the testing data set based on the sentiment lexicons of the bESD. The positive documents of the training data set are transferred into the multi-dimensional vectors based on the sentiment lexicons of the bESD, called the positive vector group of the training data set. The negative documents of the training data set are transferred into the multi-dimensional vectors based on the sentiment lexicons of the bESD, called the negative vector group of the training data set. Then, we identify one positive multi-dimensional central vector which is a central vector of the positive vector group. We calculate one negative multi-dimensional central vector which is a central vector of the negative vector group. All the multi-mensional vectors of the testing data set are clustered into either the positive polarity or the negative polarity by using the SOM with the input is all the multi-dimensional vectors of the testing

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

data set. We set an initialization of the SOM with its map in Figure 7 as follows:

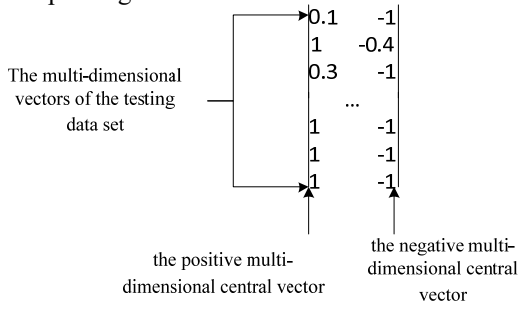

Figure 7: An initialization of the SOM – the Map

Then, after the SOM is implemented completely, we have the Map in Figure 8 as follows:

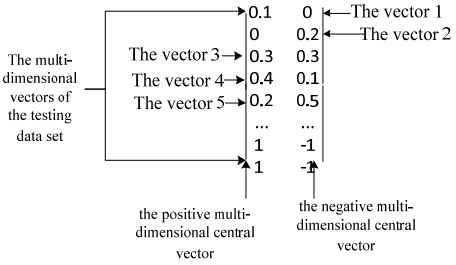

Figure 8: The final Map – the result of clustering by using the SOM

In Figure 8, we have the multi-dimensional vector 1 (0.1, 0), the multi-dimensional vector 2 (0, 0.2), the multi-dimensional vector 3 (0.3, 0.3), the multi-dimensional vector 4 (0.4, 0.1) , and the multi-dimensional vector 5 (0.2, 0.5) on the Map as follows: With the multi-dimensional vector 1 (corresponding to the document 1 of the testing data set), the column of the positive polarity is 0.1 and the column of the negative polarity is 0, Thus, the value of the column of the positive polarity is greater than the value of the column of the negative polarity, therefore this multi-dimensional vector is clustered into the positive. With the multi-dimensional vector 2 (corresponding to the document 2 of the testing data set), the column of the positive polarity is 0 and the column of the negative polarity is 0.2. Therefore, the value of the column of the positive polarity is less than the value of the column of the negative polarity, thus this multi-dimensional vector is clustered into the negative. With the multi-dimensional vector 3 (corresponding to the document 3 of the testing data set), the column of the positive polarity is 0.3 and the column of the negative polarity is 0.3. So, the value of the column of the positive polarity is as equal as the value of the column of the negative polarity, therefore this multi-dimensional vector is not clustered into both the positive and the negative. It means that this multi-dimensional vector is clustered into the neutral polarity. With the multi-dimensional vector 4 (0.4, 0.1) (corresponding to the document 4 of the testing data set), the column of the positive polarity is 0.4 and the column of the negative polarity is 0.1. Thus, the value of the column of the positive polarity is greater than the value of the column of the negative polarity, so this multi-dimensional vector is clustered into the positive. With the multi-dimensional vector 5 (corresponding to the document 5 of the testing data set), the column of the positive polarity is 0.2 and the column of the negative polarity is 0.5. Therefore, the value of the column of the positive polarity is less than the column of the negative polarity, thus this multi-dimensional vector is clustered into the negative. Finally, the sentiment classification of all the documents of the testing data set is identified completely.

We build the algorithm 1 to transfer one English document into one multi-dimensional vector according to the sentiment lexicons of the bESD in the sequential environment. The main ideas of the algorithm 1 are as follows:

Input: one English document
Output: the multi-dimensional vector
Step 1: Split the English document into many separate sentences based on "." Or "!" or "?";
Step 2: Set Multi-dimensionalVector := { } { } with n_max rows and m_max columns;
Step 3: Set i := 0;
Step 4: Each sentence in the sentences of this document, do repeat:
Step 5: Multi-dimensionalVector[i][] := {};
Step 6: Set j := 0;
Step 7: Split this sentence into the meaningful terms (meaningful words or meaningful phrases);
Step 8: Get the valence of this term based on the sentiment lexicons of the bESD;
Step 9: Add this term into Multi-dimensionalVector[i];
Step 10: Set j := j+ 1;
Step 11: End Repeat – End Step 4;
Step 12: While j is less than m_max, repeat:
Step 13: Add {0} into Multi-dimensionalVector[i];
Step 14: Set j := j+1;
Step 15: End Repeat – End Step 12;
Step 16: Set i := i+1;
Step 17: End Repeat – End Step 4;
Step 18: While i is less than n_max, repeat:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 19: Add the vector {0} into Multi-dimensionalVector;
Step 20: Set i := i+1;
Step 21: End Repeat – End Step 18;
Step 22: Return Multi-dimensionalVector;

We propose the algorithm 2 to transfer all the documents of the testing data set into the multi-dimensional vectors based on the sentiment lexicons of the bESD in the sequential environment. The main ideas of the algorithm 2 are as follows:
Input: the documents of the testing data set
Output: the multi-dimensional vectors of the testing data set
Step 1: Set TheMulti-dimensionalVectors := {}
Step 2: Each document in the documents of the testing data set, do repeat:
Step 3: OneMulti-dimensionalVector := the algorithm 4 to transfer one English document into one multi-dimensional vector according to the sentiment lexicons of the bESD in the sequential environment with the input is this document;
Step 4: Add OneMulti-dimensionalVector into TheMulti-dimensionalVectors;
Step 5: End Repeat- End Step 2;
Step 6: Return TheMulti-dimensionalVectors;

We propose the algorithm 3 to transfer all the positive documents of the training data set into all the multi-dimensional vectors based on the sentiment lexicons of the bESD, called the positive vector group of the training data set in the sequential system. The main ideas of the algorithm 3 are as follows:
Input: all the positive documents of the training data set;
Output the positive multi-dimensional vectors, called the positive vector group
Step 1: Set ThePositiveMulti-dimensionalVectors := null;
Step 2: Each document in the positive documents, repeat:
Step 3: Multi-dimensionalVector := the algorithm 4 to transfer one English document into one multi-dimensional vector according to the sentiment lexicons of the bESD in the sequential environment with the input is this document;
Step 4: Add Multi-dimensionalVector into ThePositiveMulti-dimensionalVectors;
Step 5: End Repeat – End Step 2;
Step 6: Return ThePositiveMulti-dimensionalVectors;

We implement the algorithm 4 to transfer all the negative sentences of the training data set into all the one-dimensional vectors based on the sentiment lexicons of the bESD, called the negative vector group of the training data set in the sequential environment. The main ideas of the algorithm 4 are as follows:
Input: all the negative sentences of the training data set;
Output the negative multi-dimensional vectors, called the negative vector group
Step 1: Set TheNegativeMulti-dimensionalVectors := null;
Step 2: Each document in the negative documents, repeat:
Step 3: Multi-dimensionalVector := the algorithm 4 to transfer one English document into one multi-dimensional vector according to the sentiment lexicons of the bESD in the sequential environment with the input is this document;
Step 4: Add Multi-dimensionalVector into TheNegativeMulti-dimensionalVectors;
Step 5: End Repeat – End Step 2;
Step 6: Return TheNegativeMulti-dimensionalVectors;

We build the algorithm 5 to create one positive multi-dimensional central vector from the positive vector group of the training data set in the sequential system. The main ideas of the algorithm 5 are as follows:
Input: ThePositiveOne-dimensionalVectors - the positive one-dimensional vectors, called the positive vector group
Output: one positive multi-dimensional central vector
Step 1: Set N := the number of the positive documents of the training data set;
Step 2: Set Multi-dimensionalCentralVector := null;
Step 3: For i := 0; i < n_max; i++, repeat : //rows
Step 4: Set One-dimensionalVector := null;
Step 5: Set Value := 0;
Setp 6: For j := 0; j < m_max; j++, repeat: //columns
Step 7: For k := 0; k < N; k++, repeat: //each Multi-dimensionalVector in the positive vector group
Step 8: Multi-dimensionalVector := ThePositiveOne-dimensionalVectors[k];
Step 9: Value := Value + Multi-dimensionalVector[i][j];
Step 10: End For – End Step 7;
Step 11: Add Value into One-dimensionalVector;
Step 12: End For – End Step 6;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 13: Add One-dimensionalVector into Multi-dimensionalCentralVector;
Step 14: End For – End Step 3;
Step 15: Return Multi-dimensionalCentralVector;

We propose the algorithm 6 to create one negative multi-dimensional central vector from the negative vector group of the training data set in the sequential system. The main ideas of the algorithm 6 are as follows:
Input: TheNegativeOne-dimensionalVectors - the negative one-dimensional vectors, called the negative vector group
Output: one negative multi-dimensional central vector
Step 1: Set N := the number of the negative documents of the training data set;
Step 2: Set Multi-dimensionalCentralVector := null;
Step 3: For i := 0; i < n_max; i++, repeat : //rows
Step 4: Set One-dimensionalVector := null;
Step 5: Set Value := 0;
Setp 6: For j := 0; j < m_max; j++, repeat: //columns
Step 7: For k := 0; k < N; k++, repeat: //each Multi-dimensionalVector in the negative vector group
Step 8: Multi-dimensionalVector := TheNegativeOne-dimensionalVectors[k];
Step 9: Value := Value + Multi-dimensionalVector[i][j];
Step 10: End For – End Step 7;
Step 11: Add Value into One-dimensionalVector;
Step 12: End For – End Step 6;
Step 13: Add One-dimensionalVector into Multi-dimensionalCentralVector;
Step 14: End For – End Step 3;
Step 15: Return Multi-dimensionalCentralVector;

We build the algorithm 7 to set the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the sequential system. The main ideas of the algorithm 7 are as follows:
Input: one positive multi-dimensional central vector and one negative multi-dimensional central vector ; Matrix of the SOM
Output: the Matrix of the SOM;
Step 1: Set N := the n_max sentences of one document;
Step 2: Set PositiveOne-dimensionalCentralVector := null;
Step 3: Set NegativeOne-dimensionalCentralVector := null;
Step 4: For i := 0; i < m_max; i++, repeat://columns
Step 5: Set PositiveValue := 0;
Step 6: Set NegativeValue := 0;
Step 7: For j := 0; j < N; j++, repeat://rows
Step 8: PositiveValue := PositiveValue + the positive multi-dimensional central vector [j][i];
Step 9: NegativeValue := NegativeValue + the negative multi-dimensional central vector [j][i];
Step 10: End For – End Step 7;
Step 11: PositiveValue := (PositiveValue/N);
Step 12: NegativeValue := (NegativeValue/N);
Step 13: Add PositiveValue into PositiveOne-dimensionalCentralVector;
Step 14: Add NegativeValue into NegativeOne-dimensionalCentralVector;
Step 15: End For – End Step 4;
Step 16: Set the values of PositiveOne-dimensionalCentralVector for the first column of the Maxtrix of the SOM
Step 17: Set the values of NegativeOne-dimensionalCentralVector for the second column of the Maxtrix of the SOM
Step 18: Return the Maxtrix of the SOM;

We build the algorithm 8 to cluster all the documents of the testing data set into either the positive or the negative in the sequential system by using the SOM. The main ideas of the algorithm 8 are as follows:
Input: the documents of the testing data set and the training data set
Output: positive, negative, neutral;
Step 1: TheMulti-dimensionalVectors := the algorithm 5 to transfer all the documents of the testing data set into the multi-dimensional vectors based on the sentiment lexicons of the bESD in the sequential environment with the input is the documents of the testing data set;
Step 2: the algorithm 6 to transfer all the positive documents of the training data set into all the multi-dimensional vectors according to the sentiment lexicons of the bESD, called the positive vector group of the training data set in the sequential system
Step 3: the algorithm 7 to transfer all the negative documents of the training data set into all the multi-dimensional vectors based on the sentiment lexicons of the bESD, called the negative vector group of the training data set in the sequential environment.
Step 4: the algorithm 8 to create one positive multi-dimensional central vector from the positive vector

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

group of the training data set in the sequential system
Step 5: the algorithm 9 to create one negative multi-dimensional central vector from the negative vector group of the training data set in the sequential system
Step 6: Set TheResultsOfTheSentimentClassification := {}; and the creating a basis English sentiment dictionary (bESD) in a sequential environment in (4.2.2)
Step 7: Set Matrix := {}{} with its rows are the documents of the testing data set, the 2 columns
Step 8: Set i:= 0; and N := the documents of the testing data set;
Step 9: Each i in the 2 columns -1, do repeat:
Step 10: Set j := 0;
Step 11: the algorithm 10 to set the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the sequential system.
Step 12: Set Learning rate := 0.9;
Step 13: Set R := 0;
Step 14: While stopping condition false do step 15 to 21
Step 15: For each input vector x do step 16 to 18
Step 16: For each j neuron, compute the Euclidean distance D(j)
Step 17: Find the index J such D(j) is a minimum
Step 18: For all neurons j within a specified neighbourhood of J and for all i: wji (new)= wij(old)+ learning rate * (xi - wij (old))
Step 19: Update learning rate. It is a decreasing function of the number of epochs: learning rate (t+1) = [learning rate(t)]/2;
Step 20: Reduce radius of topolofical neighbourhood at specified times
Step 21: Test stop condition. Typically this is a small value of the learning rate with which the weight updates are insignificant.
Step 22: Set count_positive := 0 and count_negative := 0;
Step 23: Each j in the N -1, do repeat:
Step 24: If Matrix[j][0] is greater than Matrix[j][1] Then OneResult := positive;
Step 25: Else If Matrix[j][0] is less than Matrix[j][1] Then OneResult := negative;
Step 26: Else: OneResult := neutral;
Step 27: Add OneResult into TheResultsOfTheSentimentClassification;
Step 28: End Repeat – End Step 23;
Step 29: Return TheResultsOfTheSentimentClassification;

4.1.2Using Self-Organizing Map Algorithm to cluster the documents of the testing data set into either the positive or the negative in a distributed system

In Figure 9, we use Self-Organizing Map Algorithm to cluster the documents of the testing data set into either the positive or the negative in a distributed environment as follows:

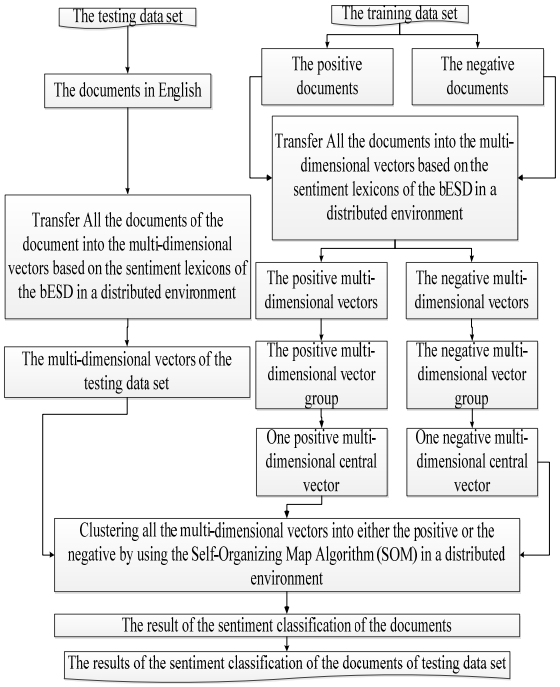

Figure 9: Overview of using Self-Organizing Map Algorithm to cluster the documents of the testing data set into either the positive or the negative in a parallel environment

In Figure 10, this section is implemented in the distributed system as follows: Based on the creating a basis English sentiment dictionary (bESD) in a distributed system in (4.2.3), the sentiment scores and the polarities of the sentiment lexicons are identified completely. All the documents of the testing data set are transferred into the multi-dimensional vectors of the testing data set based on the sentiment lexicons of the bESD. The positive documents of the training data set are transferred into the multi-dimensional vectors based on the sentiment lexicons of the bESD, called the positive vector group of the training data set. The negative documents of the training data set are transferred into the multi-dimensional vectors based on the sentiment lexicons of the bESD, called the negative vector group of the training data set. Then, we identify one positive multi-dimensional central

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

vector which is a central vector of the positive vector group. We calculate one negative multi-dimensional central vector which is a central vector of the negative vector group. All the multi-mensional vectors of the testing data set are clustered into either the positive polarity or the negative polarity by using the SOM with the input is all the multi-dimensional vectors of the testing data set. We set an initialization of the SOM with its map in Figure 10 as follows:

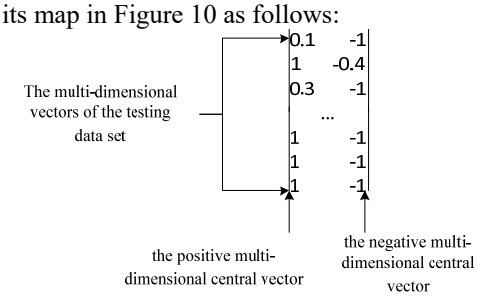

Figure 10: An initialization of the SOM – the Map

Then, after the SOM is implemented completely, we have the Map in Figure 11 as follows:

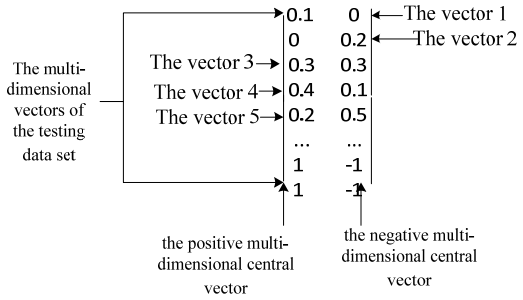

Figure 11: The final Map – the result of clustering by using the SOM

In Figure 11, we have the multi-dimensional vector 1 (0.1, 0), the multi-dimensional vector 2 (0, 0.2), the multi-dimensional vector 3 (0.3, 0.3), the multi-dimensional vector 4 (0.4, 0.1) , and the multi-dimensional vector 5 (0.2, 0.5) on the Map as follows: With the multi-dimensional vector 1 (corresponding to the document 1 of the testing data set), the column of the positive polarity is 0.1 and the column of the negative polarity is 0, Thus, the value of the column of the positive polarity is greater than the value of the column of the negative polarity, therefore this multi-dimensional vector is clustered into the positive. With the multi-dimensional vector 2 (corresponding to the document 2 of the testing data set), the column of the positive polarity is 0 and the column of the negative polarity is 0.2. Therefore, the value of the column of the positive polarity is less than the value of the column of the negative polarity, thus this multi-dimensional vector is clustered into the negative. With the multi-dimensional vector 3 (corresponding to the document 3 of the testing data set), the column of the positive polarity is 0.3 and the column of the negative polarity is 0.3. So, the value of the column of the positive polarity is as equal as the value of the column of the negative polarity, therefore this multi-dimensional vector is not clustered into both the positive and the negative. It means that this multi-dimensional vector is clustered into the neutral polarity. With the multi-dimensional vector 4 (0.4, 0.1) (corresponding to the document 4 of the testing data set), the column of the positive polarity is 0.4 and the column of the negative polarity is 0.1. Thus, the value of the column of the positive polarity is greater than the value of the column of the negative polarity, so this multi-dimensional vector is clustered into the positive. With the multi-dimensional vector 5 (corresponding to the document 5 of the testing data set), the column of the positive polarity is 0.2 and the column of the negative polarity is 0.5. Therefore, the value of the column of the positive polarity is less than the column of the negative polarity, thus this multi-dimensional vector is clustered into the negative. Finally, the sentiment classification of all the documents of the testing data set is identified completely.

In Figure 12, we transfer one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system.

In Figure 12, this stage includes two phases: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map phase is one document. The output of the Hadoop Mp is one one-dimensional vector. The input of the Hadoop Reduce is the Hadoop Map, thus, the input of the Hadoop Reduce is one one-dimensional vector. The output of the Hadoop Reduce is the multi-dimensional vector of this document.

We propose the algorithm 9 to implement the Hadoop Map phase of transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system. The main ideas of the algorithm 9 are as follows:

Input: one document

Output: one one-dimensional vector

Step 1: Input this document into the Hadoop Map in the Cloudera system.

Step 2: Split this document into the sentences;

Step 3: Each sentence in the sentences, do repeat:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 4: One-dimensionalVector := null;
Step 5: Split this sentence into the meaningful terms;
Step 6: Each term in the meaningful terms, repeat:
Step 7: Get the valence of this term based on the sentiment lexicons of the bESD;
Step 8: Add this term into One-dimensionalVector;
Step 9 End Repeat – End Step 6;
Step 10: Return this One-dimensionalVector;
Step 11: The output of the Hadoop Map is this One-dimensionalVector;
We propose the algorithm 10 to implement the Hadoop Reduce phase of transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system. The main ideas of the algorithm 10 are as follows:
Input: One-dimensionalVector - one one-dimensional vector of the Hadoop Map (the input of the Hadoop Reduce is the output of the Hadoop Map)
Output: the multi-dimensional vector of the English document – Multi-dimensionalVector;
Step 1: Receive One-dimensionalVector;
Step 2: Add this One-dimensionalVector into One-dimensionalVector;
Step 3: Return Multi-dimensionalVector;

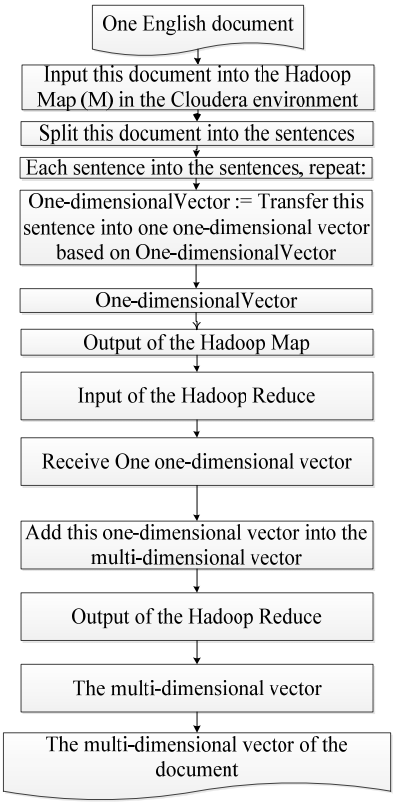

Figure 12: Overview of transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system

In Figure 13, we transfer the documents of the testing data set into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system as follows:

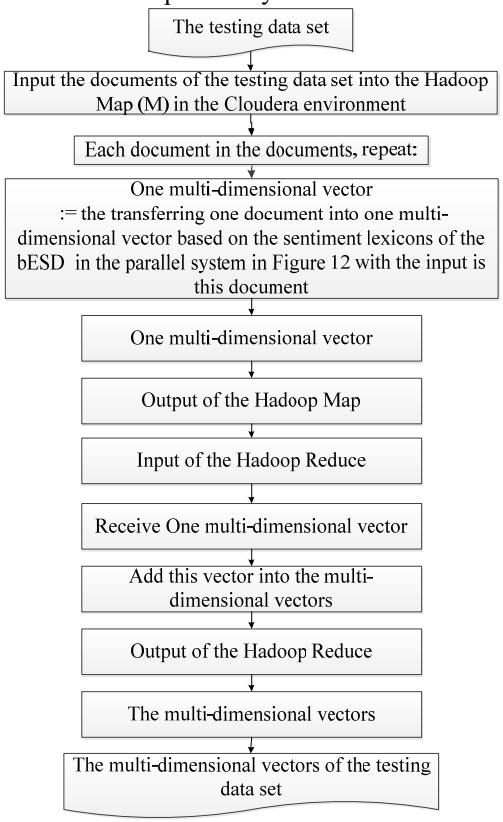

Figure 13: Overview of transferring the documents of the testing data set into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system

In Figure 14, this stage includes two phases: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map phase is the documents of the testing data set. The output of the Hadoop Mp is one multi-dimensional vector. The input of the Hadoop Reduce is the Hadoop Map, thus, the input of the Hadoop Reduce is one multi-dimensional vector. The output of the Hadoop Reduce is the multi-dimensional vectors of the testing data set
We propose the algorithm 11 to implement the Hadoop Map phase of transferring one document into the one-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system. The main ideas of the algorithm 11 are as follows:
Input: the documents of the testing data set

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Output: one multi-dimensional vector (corresponding to one document)
Step 1: Input the documents of the testing data set into the Hadoop Map in the Cloudera system.
Step 3: Each document in the documents of the testing data set, do repeat:
Step 4: the multi-dimensional vector := the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system in Figure 12 with the input is this document;
Step 5: Return this multi-dimensional vector;
Step 6: The output of the Hadoop Map is this multi-dimensional vector;
We propose the algorithm 12 to implement the Hadoop Reduce phase of transferring one document into the one-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system. The main ideas of the algorithm 12 are as follows:
Input: one multi-dimensional vector of the Hadoop Map (the input of the Hadoop Reduce is the output of the Hadoop Map)
Output: the multi-dimensional vectors of the English documents of the testing data set
Step 1: Receive one multi-dimensional vector of the Hadoop Map
Step 2: Add this multi-dimensional vector into the multi-dimensional vectors of the testing data set;
Step 3: Return the multi-dimensional vectors of the testing data set;

In Figure 14, we transfer the positive documents of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

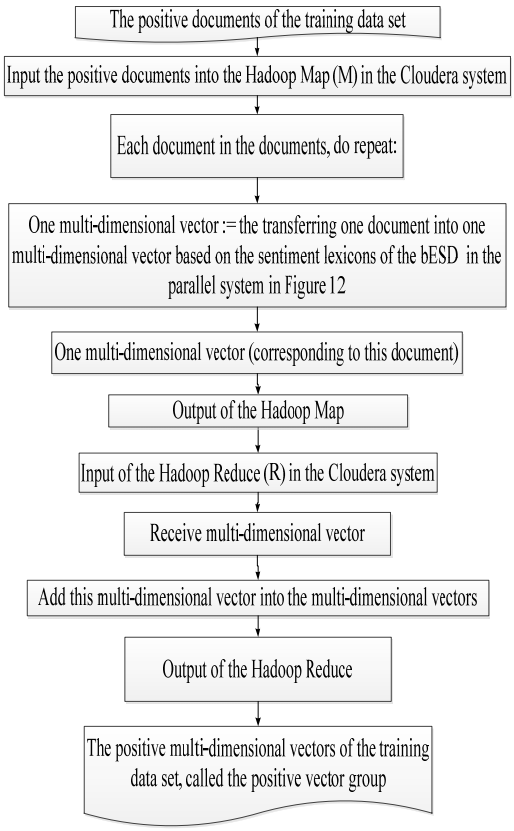

Figure 14: Overview of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

In Figure 14, the stage includes two phases: the Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase is the positive documents of the training data set. The output of the Hadoop Map phase is one multi-dimensional vector (corresponding to one document of the positive documents of the training data set). The input of the Hadoop Redude phase is the output of the Hadoop Map phase, thus, the input of the Hadoop Reduce phase is one multi-dimensional vector (corresponding to one document of the positive documents of the training data set). The output of the Hadoop Reduce phase is the positive multi-dimensional vectors, called the positive vector group (corresponding to the positive documents of the training data set)
We propose the algorithm 13 to perform the Hadoop Map phase of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

system. The main ideas of the algorithm 13 are as follows:
Input: the positive documents of the training data set
Output: one multi-dimensional vector (corresponding to one document of the positive documents of the training data set)
Step 1: Input the positive documents into the Hadoop Map in the Cloudera system.
Step 2: Each document in the documents, do repeat:
Step 3: MultiDimentionalVector := the transferring one document into one multi-dimensional vector based the sentiment lexicons of the bESD in the parallel system in Figure 12
Step 4: Return MultiDimentionalVector ;
We propose the algorithm 14 to implement the Hadoop Reduce phase of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system. The main ideas of the algorithm 14 are as follows:
Input: one multi-dimensional vector (corresponding to one document of the positive documents of the training data set)
Output: the positive multi-dimensional vectors, called the positive vector group (corresponding to the positive documents of the training data set)
Step 1: Receive one multi-dimensional vector;
Step 2: Add this multi-dimensional vector into PositiveVectorGroup;
Step 3: Return PositiveVectorGroup - the positive multi-dimensional vectors, called the positive vector group (corresponding to the positive documents of the training data set);

In Figure 15, we transfer the negative documents of the training data set into the negative multi-dimensional vectors (called the negative vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

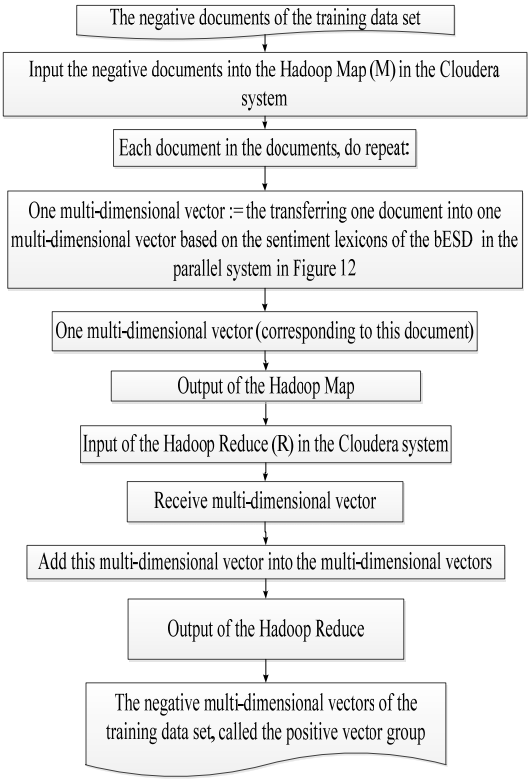

Figure 15: Overview of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

In Figure 15, the stage includes two phases: the Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase is the negative documents of the training data set. The output of the Hadoop Map phase is one multi-dimensional vector (corresponding to one document of the negative documents of the training data set). The input of the Hadoop Redude phase is the output of the Hadoop Map phase, thus, the input of the Hadoop Reduce phase is one multi-dimensional vector (corresponding to one document of the negative documents of the training data set). The output of the Hadoop Reduce phase is the negative multi-dimensional vectors, called the negative vector group (corresponding to the negative documents of the training data set)
We build the algorithm 15 to perform the Hadoop Map phase of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

system. The main ideas of the algorithm 15 are as follows:
Input: the negative documents of the training data set
Output: one multi-dimensional vector (corresponding to one document of the negative documents of the training data set)
Step 1: Input the negative documents into the Hadoop Map in the Cloudera system.
Step 2: Each document in the documents, do repeat:
Step 3: MultiDimentionalVector := the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system in Figure 12
Step 4: Return MultiDimentionalVector ;
We propose the algorithm 16 to implement the Hadoop Reduce phase of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system. The main ideas of the algorithm 16 are as follows:
Input: one multi-dimensional vector (corresponding to one document of the negative documents of the training data set)
Output: the negative multi-dimensional vectors, called the negative vector group (corresponding to the negative documents of the training data set)
Step 1: Receive one multi-dimensional vector;
Step 2: Add this multi-dimensional vector into NegativeVectorGroup;
Step 3: Return NegativeVectorGroup - the megative multi-dimensional vectors, called the negative vector group (corresponding to the negative documents of the training data set);

In Figure 16, we create one positive multi-dimensional central vector from the positive vector group of the training data set in the distributed system

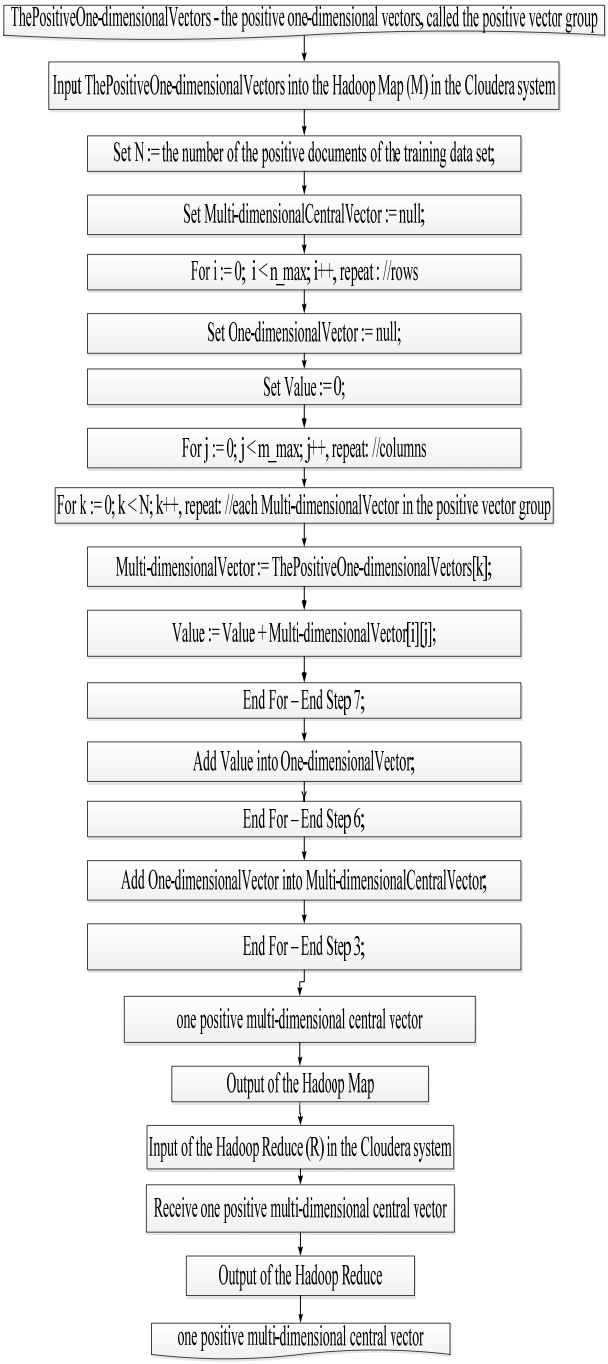

Figure 16: Overview creating one positive multi-dimensional central vector from the positive vector group of the training data set in the distributed system

In Figure 16, the stage includes two phases: the Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase is ThePositiveOne-dimensionalVectors - the positive

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

one-dimensional vectors, called the positive vector group. The output of the Hadoop Map phase is one positive multi-dimensional central vector. The input of the Hadoop Redude phase is the output of the Hadoop Map phase, thus, the input of the Hadoop Reduce phase is one positive multi-dimensional central vector. The output of the Hadoop Reduce phase is one positive multi-dimensional central vector

We build the algorithm 17 to implement the Hadoop Map phase of creating one positive multi-dimensional central vector from the positive vector group of the training data set in the distributed system. The main ideas of the algorithm 17 are as follows:

Input: ThePositiveOne-dimensionalVectors - the positive one-dimensional vectors, called the positive vector group

Output: one positive multi-dimensional central vector

Step 1: Set N := the number of the positive documents of the training data set;

Step 2: Set Multi-dimensionalCentralVector := null;

Step 3: For i := 0; i < n_max; i++, repeat : //rows

Step 4: Set One-dimensionalVector := null;

Step 5: Set Value := 0;

Step 6: For j := 0; j < m_max; j++, repeat: //columns

Step 7: For k := 0; k < N; k++, repeat: //each Multi-dimensionalVector in the positive vector group

Step 8: Multi-dimensionalVector := ThePositiveOne-dimensionalVectors[k];

Step 9: Value := Value + Multi-dimensionalVector[i][j];

Step 10: End For – End Step 7;

Step 11: Add Value into One-dimensionalVector;

Step 12: End For – End Step 6;

Step 13: Add One-dimensionalVector into Multi-dimensionalCentralVector;

Step 14: End For – End Step 3;

Step 15: Return Multi-dimensionalCentralVector;

Step 16: The output of the Hadoop Map is Multi-dimensionalCentralVector;

We build the algorithm 18 to perform the Hadoop Reduce phase of creating one positive multi-dimensional central vector from the positive vector group of the training data set in the distributed system. The main ideas of the algorithm 18 are as follows:

Input: one positive multi-dimensional central vector – the output of the Hadoop Map

Output: one positive multi-dimensional central vector

Step 1: Receive one positive multi-dimensional central vector;

Step 2: Return one positive multi-dimensional central vector;

In Figure 17, we create one negative multi-dimensional central vector from the negative vector group of the training data set in the distributed system

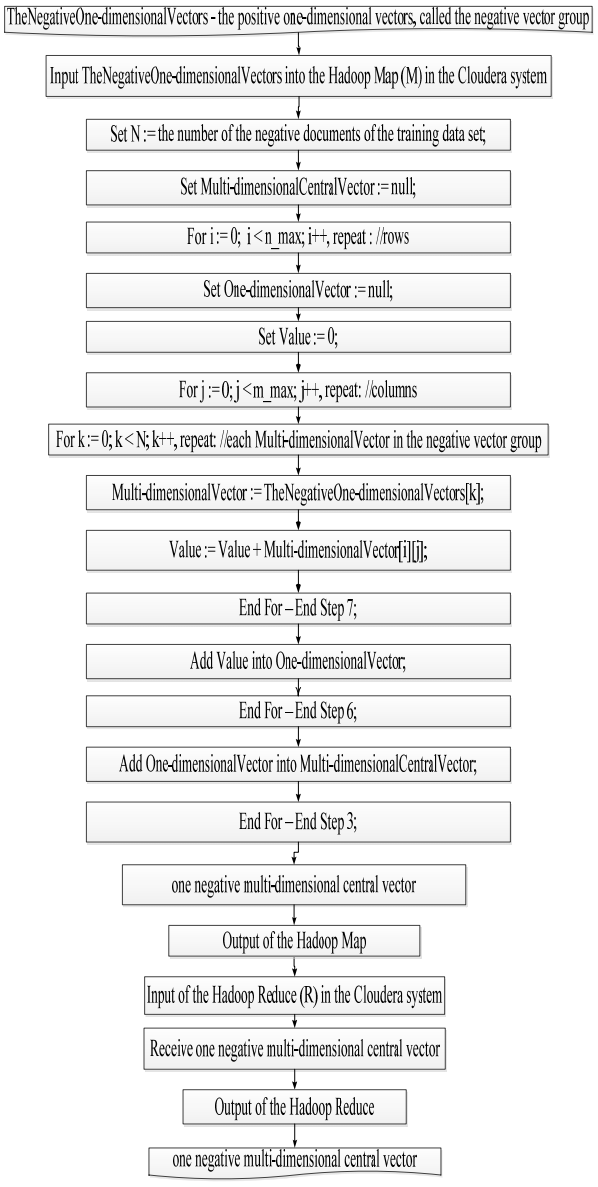

Figure 17: Overview of creating one negative multi-dimensional central vector from the negative vector group of the training data set in the distributed system

In Figure 17, the stage includes two phases: the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase is TheNegativeOne-dimensionalVectors - the positive one-dimensional vectors, called the negative vector group. The output of the Hadoop Map phase is one negative multi-dimensional central vector. The input of the Hadoop Redude phase is the output of the Hadoop Map phase, thus, the input of the Hadoop Reduce phase is one negative multi-dimensional central vector. The output of the Hadoop Reduce phase is one negative multi-dimensional central vector

We build the algorithm 19 to implement the Hadoop Map phase of creating one negative multi-dimensional central vector from the negative vector group of the training data set in the distributed system. The main ideas of the algorithm 19 are as follows:

Input: TheNegativeOne-dimensionalVectors - the negative one-dimensional vectors, called the negative vector group

Output: one negative multi-dimensional central vector

Step 1: Set N := the number of the negative documents of the training data set;

Step 2: Set Multi-dimensionalCentralVector := null;

Step 3: For i := 0; i < n_max; i++, repeat : //rows

Step 4: Set One-dimensionalVector := null;

Step 5: Set Value := 0;

Setp 6: For j := 0; j < m_max; j++, repeat: //columns

Step 7: For k := 0; k < N; k++, repeat: //each Multi-dimensionalVector in the negative vector group

Step 8: Multi-dimensionalVector := TheNegativeOne-dimensionalVectors[k];

Step 9: Value := Value + Multi-dimensionalVector[i][j];

Step 10: End For – End Step 7;

Step 11: Add Value into One-dimensionalVector;

Step 12: End For – End Step 6;

Step 13: Add One-dimensionalVector into Multi-dimensionalCentralVector;

Step 14: End For – End Step 3;

Step 15: Return Multi-dimensionalCentralVector;

Step 16: The output of the Hadoop Map is Multi-dimensionalCentralVector;

We build the algorithm 20 to perform the Hadoop Reduce phase of creating one negative multi-dimensional central vector from the negative vector group of the training data set in the distributed system. The main ideas of the algorithm 20 are as follows:

Input: one negative multi-dimensional central vector – the output of the Hadoop Map

Output: one negative multi-dimensional central vector

Step 1: Receive one negative multi-dimensional central vector;

Step 2: Return one negative multi-dimensional central vector;

In Figure 18, we set the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the distributed system

In Figure 18, the stage includes two phases: the Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase is one positive multi-dimensional central vector and one negative multi-dimensional central vector; Matrix of the SOM. The output of the Hadoop Map phase is the Matrix of the SOM. The input of the Hadoop Redude phase is the output of the Hadoop Map phase, thus, the input of the Hadoop Reduce phase is the Matrix of the SOM. The output of the Hadoop Reduce phase is the Matrix of the SOM.

We build the algorithm 21 to perform the Hadoop Map phase of setting the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the distributed system. The main ideas of the algorithm 21 are as follows:

Input: one positive multi-dimensional central vector and one negative multi-dimensional central vector ; Matrix of the SOM

Output: the Matrix of the SOM;

Step 0: Input one positive multi-dimensional central vector and one negative multi-dimensional central vector ; Matrix of the SOM into the Hadoop Map in the Cloudera system;

Step 1: Set N := the n_max sentences of one document;

Step 2: Set PositiveOne-dimensionalCentralVector := null;

Step 3: Set NegativeOne-dimensionalCentralVector := null;

Step 4: For i := 0; i < m_max; i++, repeat://columns

Step 5: Set PositiveValue := 0;

Step 6: Set NegativeValue := 0;

Step 7: For j := 0; j < N; j++, repeat://rows

Step 8: PositiveValue := PositiveValue + the positive multi-dimensional central vector [j][i];

Step 9: NegativeValue := NegativeValue + the negative multi-dimensional central vector [j][i];
Step 10: End For – End Step 7;
Step 11: PositiveValue := (PositiveValue/N);
Step 12: NegativeValue := (NegativeValue/N);
Step 13: Add PositiveValue into PositiveOne-dimensionalCentralVector;
Step 14: Add NegativeValue into NegativeOne-dimensionalCentralVector;
Step 15: End For – End Step 4;
Step 16: Set the values of PositiveOne-dimensionalCentralVector for the first column of the Maxtrix of the SOM
Step 17: Set the values of NegativeOne-dimensionalCentralVector for the second column of the Maxtrix of the SOM
Step 18: Return the Maxtrix of the SOM;

We build the algorithm 22 to implement the Hadoop Reduce phase of setting the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the parallel system. The main ideas of the algorithm 22 are as follows:

Input: the Matrix of the SOM – the output of the Hadoop Map;
Output: the Matrix of the SOM;
Step 1:Receive the Matrix of the SOM;
Step 2: Return the Matrix of the SOM;

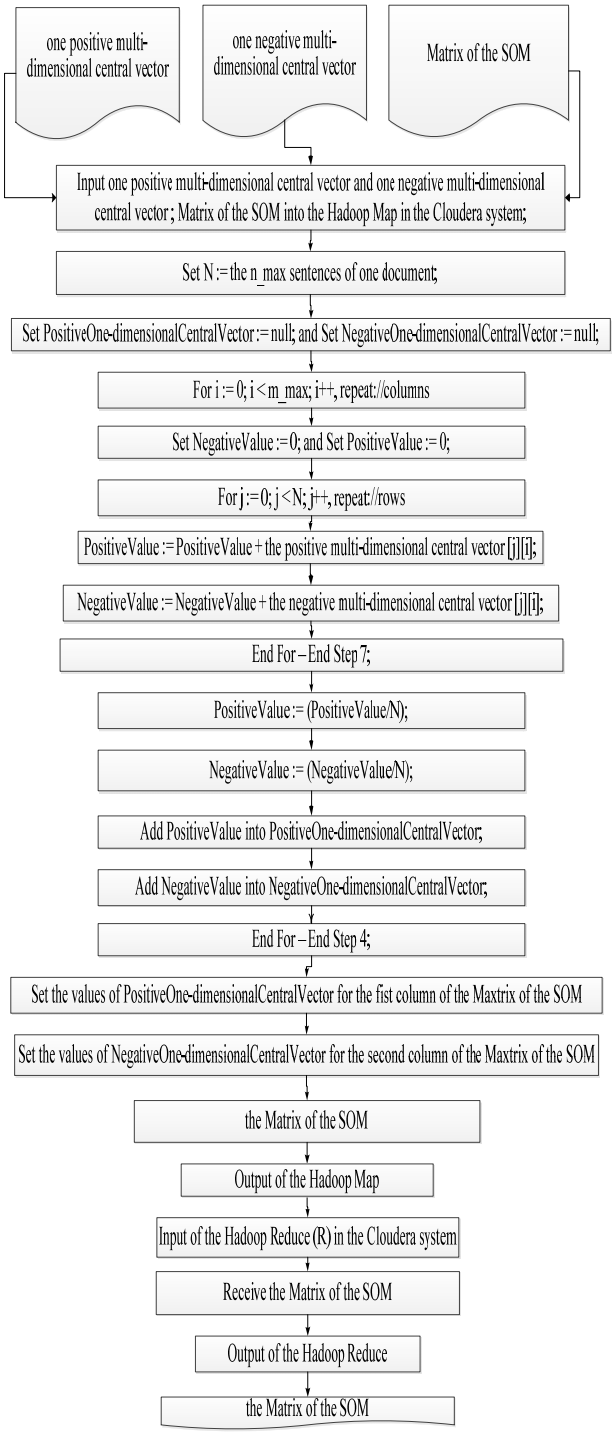

Figure 18: Overview of setting the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the parallel system

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In Figure 19, we use the Self-Organizing Map Algorithm (SOM) to cluster the documents of the testing data set into either the positive or the negative in the distributed system.

In Figure 19, this stage comprises two phases: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map is the documents of the testing data set. The output of the Hadoop Map is the result of the sentiment classification of one document. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is the result of the sentiment classification of one document. The output of the Hadoop Reduce is the results of the sentiment classification of the documents of the testing data set.

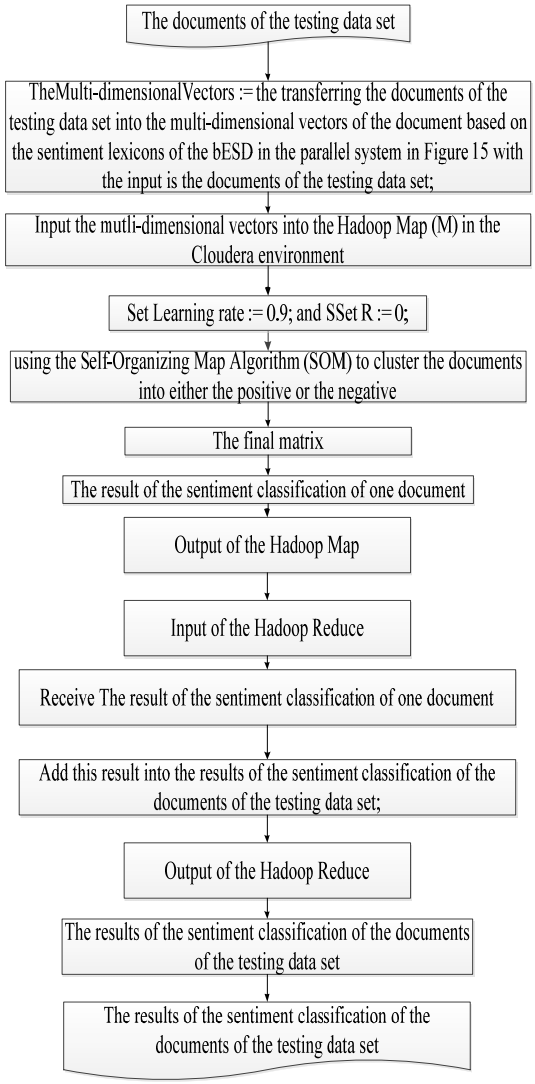

Figure 19: Overview of using the Self-Organizing Map Algorithm (SOM) to cluster the documents into either the positive or the negative in the distributed system.

We propose the algorithm 23 to perform the Hadoop Map phase of using the Self-Organizing Map Algorithm (SOM) to cluster the documents into either the positive or the negative in the distributed system. The main ideas of the algorithm 23 are as follows:

Input: the documents of the testing data set and the training data set

Output: positive, negative, neutral;

Step 0: the creating a basis English sentiment dictionary (bESD) in a distributed system in (4.2.3);

Step 1: TheMulti-dimensionalVectors := the transferring the documents of the testing data set into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system in Figure 13

Step 2: the transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system in Figure 14

Step 3: the transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system in Figure 15

Step 4: the creating one positive multi-dimensional central vector from the positive vector group of the training data set in the distributed system in Figure 16

Step 5: the creating one negative multi-dimensional central vector from the negative vector group of the training data set in the distributed system in Figure 17

Step 6: the setting the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the distributed system in Figure 18

Step 7: Input TheMulti-dimensionalVectors and the Matrix into the Hadoop Map in the Cloudera system;

Step 8: Set N := the documents of the testing data set;

Step 9: Set Learning rate := 0.9;

Step 10: Set R := 0;

Step 11: While stopping condition false do step 12 to 18

Step 12: For each input vector x do step 13 to 15

Step 13: For each j neuron, compute the Euclidean distance D(j)

Step 14: Find the index J such D(j) is a minimum

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 15: For all neurons j within a specified neighbourhood of J and for all i: wji (new)= wij(old)+ learning rate * (xi - wij (old))
Step 16: Update learning rate. It is a decreasing function of the number of epochs: learning rate (t+1) = [learning rate(t)]/2;
Step 17: Reduce radius of topolofical neighbourhood at specified times
Step 18: Test stop condition. Typically this is a small value of the learning rate with which the weight updates are insignificant.
Step 19: Set count_positive := 0 and count_negative := 0;
Step 20: Each j in the N -1, do repeat:
Step 21: If Matrix[j][0] is greater than Matrix[j][1] Then OneResult := positive;
Step 22: Else If Matrix[j][0] is less than Matrix[j][1] Then OneResult := negative;
Step 23: Else: OneResult := neutral;
Step 24: Return OneResult;
Step 25: The output of the Hadoop map is the OneResult;
We propose the algorithm 24 to implement the Hadoop Reduce phase of the using the Self-Organizing Map Algorithm (SOM) to cluster the documents into either the positive or the negative in the distributed system. The main ideas of the algorithm 24 are as follows:
Input: OneResult - the result of the sentiment classification of one document (the input of the Hadoop Reduce is the output of the Hadoop Map)
Output: the results of the sentiment classification of the documents of the testing data set;
Step 1: Receive OneResult - the result of the sentiment classification of one document
Step 2: Add this OneResult into the results of the sentiment classification of the documents of the testing data set;
Step 3: Return the results of the sentiment classification of the documents of the testing data set;

4.2 Creating the sentiment lexicons of the bESD

The section includes three parts. The first sub-section of this section is to identify a sentiment value of one word (or one phrase) in English in the sub-section (4.2.1). The second part of this section is to create a basis English sentiment dictionary (bESD) in a sequential system in the sub-section (4.2.2). The third sub-section of this section is to create a basis English sentiment dictionary (bESD) in a parallel environment in the sub-section (4.2.3).

4.2.1 Calculating a valence of one word (or one phrase) in English

In this part, we calculate the valence and the polarity of one English word (or phrase) by using the YIIM through a Google search engine with AND operator and OR operator, as the following diagram in Figure 20 below shows.

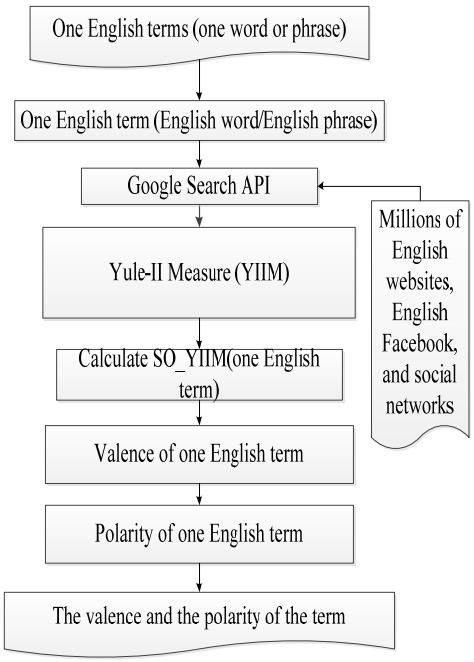

Figure 20: Overview of identifying the valence and the polarity of one term in English using an Yule-II coefficient (YIIM)

According to [20-34], Pointwise Mutual Information (PMI) between two words wi and wj has the equation

$$PMI(wi, wj) = log_2(\frac{P(wi, wj)}{P(wi)xP(wj)}) \quad (1)$$

and SO (sentiment orientation) of word wi has the equation

$$SO\ (wi) = PMI(wi, positive) - PMI(wi, negative) \quad (2)$$

In [20-27] the positive and the negative of Eq. (2) in English are: positive = {good, nice, excellent, positive, fortunate, correct, superior} and negative = {bad, nasty, poor, negative, unfortunate, wrong, inferior}.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The AltaVista search engine is used in the PMI equations of [21, 22, 24] and the Google search engine is used in the PMI equations of [23, 25, 27]. Besides, [23] also uses German, [24] also uses Medonian, [25] also uses Arabic, [26] also uses Chinese, and [27] also uses Spanish. In addition, the Bing search engine is also used in [25].

With [28-31], the PMI equations are used in Chinese, not English, and Tibetan is also added in [9]. About the search engine, the AltaVista search engine is used in [30] and [31] and uses three search engines, such as the Google search engine, the Yahoo search engine and the Baidu search engine. The PMI equations are also used in Japanese with the Google search engine in [32]. [33] and [34] also use the PMI equations and Jaccard equations with the Google search engine in English.

According to [33-41], Jaccard between two words wi and wj has the equations

$$Jaccard(wi, wj) = J(wi, wj) = \frac{|wi \cap wj|}{|wi \cup wj|} \quad (3)$$

and other type of the Jaccard equation between two words wi and wj has the equation

$$\text{Jaccard}(\text{wi}, \text{wj}) = \text{J}(\text{wi}, \text{wj}) = \text{sim}(\text{wi}, \text{wj}) = \frac{\text{F}(\text{wi}, \text{wj})}{\text{F}(\text{wi}) + \text{F}(\text{wj}) - \text{F}(\text{wi}, \text{wj})} \quad (4)$$

and SO (sentiment orientation) of word wi has the equation

$$\text{SO}(\text{wi}) = \sum \text{Sim}(\text{wi}, \text{positive}) - \sum \text{Sim}(\text{wi}, \text{positive}) \quad (5)$$

In [33-40] the positive and the negative of Eq. (5) in English are: positive = {good, nice, excellent, positive, fortunate, correct, superior} and negative = {bad, nasty, poor, negative, unfortunate, wrong, inferior}.

The Jaccard equations with the Google search engine in English are used in [33, 34, 36]. [16] and [40] use the Jaccard equations in English. [39] and [41] use the Jaccard equations in Chinese. [37] uses the Jaccard equations in Arabic. The Jaccard equations with the Chinese search engine in Chinese are used in [38].

The authors in [47] used the Ochiai Measure through the Google search engine with AND operator and OR operator to calculate the sentiment values of the words in Vietnamese. The authors in [48] used the Cosine Measure through the Google search engine with AND operator and OR operator to identify the sentiment scores of the words in English. The authors in [49] used the Sorensen Coefficient through the Google search engine with AND operator and OR operator to calculate the sentiment values of the words in English. The authors in [50] used the Jaccard Measure through the Google search engine with AND operator and OR operator to calculate the sentiment values of the words in Vietnamese. The authors in [51] used the Tanimoto Coefficient through the Google search engine with AND operator and OR operator to identify the sentiment scores of the words in English

With the above proofs, we have this: PMI is used with AltaVista in English, Chinese, and Japanese with the Google in English; Jaccard is used with the Google in English, Chinese, and Vietnamese. The Ochiai is used with the Google in Vietnamese. The Cosine and Sorensen are used with the Google in English.

According to [20-51], PMI, Jaccard, Cosine, Ochiai, Sorensen, Tanimoto, and Yule-II coefficient (YIIM) are the similarity measures between two words, and they can perform the same functions and with the same characteristics; so YIIM is used in calculating the valence of the words. In addition, we prove that YIIM can be used in identifying the valence of the English word through the Google search with the AND operator and OR operator.

With the Yule-II coefficient (YIIM) in [58-63], we have the equation of the YIIM:

$$\text{Yule} - \text{II Coefficient}(a, b) = \text{Yule} - \text{II Measure}(a, b) = \text{YIIM}(a, b) = \frac{(a \cap b) * (\neg a \cap \neg b) - (\neg a \cap b) * (a \cap \neg b)}{\sqrt{[(a \cap b) + (\neg a \cap b)] * [(a \cap b) + (a \cap \neg b)] * [(\neg a \cap b) + (\neg a \cap \neg b)] * [(a \cap \neg b) + (\neg a \cap \neg b)]}} \quad (6)$$

with a and b are the vectors.

From the eq. (1), (2), (3), (4), (5), (6), we propose many new equations of the YIIM to calculate the valence and the polarity of the English words (or the English phrases) through the Google search engine as the following equations below.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In eq. (6), when a has only one element, a is a word. When b has only one element, b is a word. In eq. (6), a is replaced by w1 and b is replaced by w2.

$$\begin{aligned}\text{Yule} - \text{II Measure}(w1, w2) &= \text{Yule} \\ &- \text{II Coefficient}(w1, w2) = \\ \text{YIIM}(w1, w2) &= \frac{B}{\sqrt{A}} \quad (7)\end{aligned}$$

with

$$\begin{aligned}B = {} & P(w1, w2) * P(\neg w1, \neg w2) - P(\neg w1, w2) \\ & * P(w1, \neg w2) \\ A = {} & [P(w1, w2) + P(\neg w1, w2)] \\ & * [P(w1, w2) + P(w1, \neg w2)] \\ & * [P(\neg w1, w2) \\ & + P(\neg w1, \neg w2)] \\ & * [P(w1, \neg w2) \\ & + P(\neg w1, \neg w2)]\end{aligned}$$

Eq. (7) is similar to eq. (1). In eq. (2), eq. (1) is replaced by eq. (7). We have eq. (8)

$$\begin{aligned}&\text{Valence}(w) = \text{SO\_YIIM}(w) \\ &= \text{YIIM}(w, \text{positive\_query}) \\ &- \text{YIIM}(w, \text{negative\_query}) \quad (8)\end{aligned}$$

In eq. (7), w1 is replaced by w and w2 is replaced by position_query. We have eq. (9). Eq. (9) is as follows:

$$\text{YIIM}(w, \text{positive\_query}) = \frac{B9}{\sqrt{A9}} \quad (9)$$

with

$$\begin{aligned}B9 = {} & P(w, \text{positive\_query}) \\ & * P(\neg w, \neg \text{positive\_query}) \\ & - P(\neg w, \text{positive\_query}) \\ & * P(w, \neg \text{positive\_query}) \\ A9 = {} & [P(w, \text{positive\_query}) \\ & + P(\neg w, \text{positive\_query})] \\ & * [P(w, \text{positive\_query}) \\ & + P(w, \neg \text{positive\_query})] \\ & * [P(\neg w, \text{positive\_query}) \\ & + P(\neg w, \neg \text{positive\_query})] \\ & * [P(w, \neg \text{positive\_query}) \\ & + P(\neg w, \neg \text{positive\_query})]\end{aligned}$$

In eq. (7), w1 is replaced by w and w2 is replaced by negative_query. We have eq. (10). Eq. (10) is as follows:

$$\text{YIIM}(w, \text{negative\_query}) = \frac{B10}{\sqrt{A10}} \quad (10)$$

with

$$\begin{aligned}B10 = {} & P(w, \text{negative\_query}) \\ & * P(\neg w, \neg \text{negative\_query}) \\ & - P(\neg w, \text{negative\_query}) \\ & * P(w, \neg \text{negative\_query}) \\ A10 = {} & [P(w, \text{negative\_query}) \\ & + P(\neg w, \text{negative\_query})] \\ & * [P(w, \text{negative\_query}) \\ & + P(w, \neg \text{negative\_query})] \\ & * [P(\neg w, \text{negative\_query}) \\ & + P(\neg w, \neg \text{negative\_query})] \\ & * [P(w, \neg \text{negative\_query}) \\ & + P(\neg w, \neg \text{negative\_query})]\end{aligned}$$

with:

1)w, w1, w2 : are the English words (or the English phrases)

2)P(w1, w2): number of returned results in Google search by keyword (w1 and w2). We use the Google Search API to get the number of returned results in search online Google by keyword (w1 and w2).

3)P(w1): number of returned results in Google search by keyword w1. We use the Google Search API to get the number of returned results in search online Google by keyword w1.

4)P(w2): number of returned results in Google search by keyword w2. We use the Google Search API to get the number of returned results in search online Google by keyword w2.

5)Valence(W) = SO_YIIM(w): valence of English word (or English phrase) w; is SO of word (or phrase) by using the Yule-II coefficient (YIIM)

6)positive_query: { active or good or positive or beautiful or strong or nice or excellent or fortunate or correct or superior } with the positive query is the a group of the positive English words.

7)negative_query: { passive or bad or negative or ugly or week or nasty or poor or unfortunate or wrong or inferior }
with the negative_query is the a group of the negative English words.

8)P(w, positive_query): number of returned results in Google search by keyword (positive_query and w). We use the Google Search API to get the number of returned results in search online Google by keyword (positive_query and w)

9)P(w, negative_query): number of returned results in Google search by keyword (negative_query and w). We use the Google Search API to get the number of returned results in search online Google

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

by keyword (negative_query and w)
10)P(w): number of returned results in Google search by keyword w. We use the Google Search API to get the number of returned results in search online Google by keyword w
11)P(¬w,positive_query): number of returned results in Google search by keyword ((not w) and positive_query). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and positive_query).
12)P(w, ¬positive_query): number of returned results in the Google search by keyword (w and (not (positive_query))). We use the Google Search API to get the number of returned results in search online Google by keyword (w and [not (positive_query)]).
13)P(¬w, ¬positive_query): number of returned results in the Google search by keyword (w and (not (positive_query))). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and [not (positive_query)]).
14)P(¬w,negative_query): number of returned results in Google search by keyword ((not w) and negative_query). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and negative_query).
15)P(w,¬negative_query): number of returned results in the Google search by keyword (w and (not (negative_query))). We use the Google Search API to get the number of returned results in search online Google by keyword (w and (not (negative_query))).
16)P(¬w,¬negative_query): number of returned results in the Google search by keyword (w and (not (negative_query))). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and (not (negative_query))).

As like Cosine, Ochiai, Sorensen, Tanimoto, PMI and Jaccard about calculating the valence (score) of the word, we identify the valence (score) of the English word w based on both the proximity of positive_query with w and the remote of positive_query with w; and the proximity of negative_query with w and the remote of negative_query with w.The English word w is the nearest of positive_query if YIIM (w, positive_query) is as equal as 1.The English word w is the farthest of positive_query if YIIM(w, positive_query) is as equal as 0. The English word w belongs to positive_query being the positive group of the English words if YIIM(w, positive_query) > 0 and YIIM(w, positive_query) ≤ 1.The English word w is the nearest of negative_query if YIIM(w, negative_query) is as equal as 1. The English word w is the farthest of negative_query if YIIM(w, negative_query) is as equal as 0. The English word w belongs to negative_query being the negative group of the English words if YIIM(w, negative_query) > 0 and YIIM(w, negative_query) ≤ 1. So, the valence of the English word w is the value of YIIM(w, positive_query) substrYIIMting the value of YIIM(w, negative_query) and the eq. (8) is the equation of identifying the valence of the English word w.
We have the information about YIIM as follows:
1)YIIM(w, positive_query) ≥ 0 and YIIM(w, positive_query) ≤ 1.
2)YIIM(w, negative_query) ≥ 0 and YIIM (w, negative_query) ≤ 1
3)If YIIM (w, positive_query) = 0 and YIIM (w, negative_query) = 0 then SO_YIIM (w) = 0.
4)If YIIM (w, positive_query) = 1 and YIIM (w, negative_query) = 0 then SO_YIIM (w) = 0.
5)If YIIM (w, positive_query) = 0 and YIIM (w, negative_query) = 1 then SO_YIIM (w) = -1.
6)If YIIM (w, positive_query) = 1 and YIIM (w, negative_query) = 1 then SO_YIIM(w) = 0.
So, SO_YIIM (w) ≥ -1 and SO_YIIM (w) ≤ 1.
The polarity of the English word w is positive polarity If SO_YIIM (w) > 0. The polarity of the English word w is negative polarity if SO_YIIM (w) < 0. The polarity of the English word w is neutral polarity if SO_YIIM (w) = 0. In addition, the semantic value of the English word w is SO_YIIM (w).
We calculate the valence and the polarity of the English word or phrase w using a training corpus of approximately one hundred billion English words — the subset of the English Web that is indexed by the Google search engine on the internet. AltaVista was chosen because it has a NEAR operator. The AltaVista NEAR operator limits the search to documents that contain the words within ten words of one another, in either order. We use the Google search engine which does not have a NEAR operator; but the Google search engine can use the AND operator and the OR operator. The result of calculating the valence w (English word) is similar to the result of calculating valence w by using AltaVista. However, AltaVista is no longer.
In summary, by using eq. (8), eq. (9), and eq. (10), we identify the valence and the polarity of one word (or one phrase) in English by using the SC through the Google search engine with AND operator and OR operator.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In Table 10, we present the comparisons of our model's results with the works related to [20-51].
The comparisons of our model's advantages and disadvantages with the works related to [20-51] are shown in Table 11.
In Table 12, we show the comparisons of our model's results with the works related to the YULE-II coefficient (MC) in [58-63]
The comparisons of our model's benefits and drawbacks with the studies related to the YULE-II coefficient (MC) in [58-63] are displayed in Table 13.

4.2.2 Creating a basis English sentiment dictionary (bESD) in a sequential environment

According to [52-57], we have at least 55,000 English terms, including nouns, verbs, adjectives, etc. In this part, we calculate the valence and the polarity of the English words or phrases for our basis English sentiment dictionary (bESD) by using the YIIM in a sequential system, as the following diagram in Figure 21 below shows.

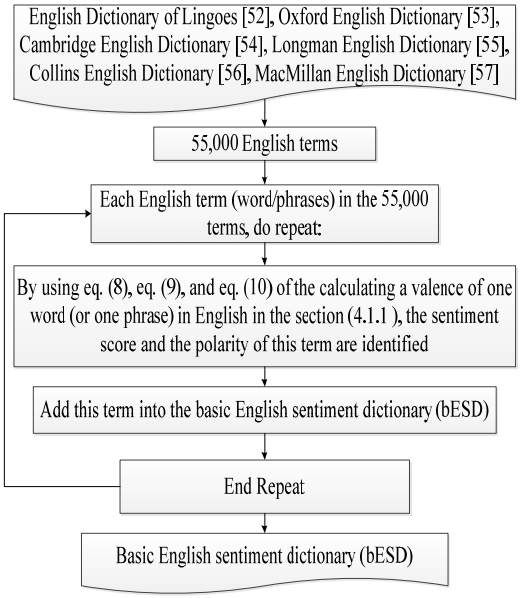

Figure 21: Overview of creating a basis English sentiment dictionary (bESD) in a sequential environment

We proposed the algorithm 25 to perform a basis English sentiment dictionary (bESD) in a sequential environment. The main ideas of the algorithm 25 are as follows:
Input: the 55,000 English terms; the Google search engine
Output: a basis English sentiment dictionary (bESD)
Step 1: Each term in the 55,000 terms, do repeat:
Step 2: By using eq. (8), eq. (9), and eq. (10) of the calculating a valence of one word (or one phrase) in English in the section (4.1.1), the sentiment score and the polarity of this term are identified. The valence and the polarity are calculated by using the MC through the Google search engine with AND operator and OR operator.
Step 3: Add this term into the basis English sentiment dictionary (bESD);
Step 4: End Repeat – End Step 1;
Step 5: Return bESD;
Our basis English sentiment dictionary (bESD) has more 55,000 English words (or English phrases) and bESD is stored in Microsoft SQL Server 2008 R2.

4.2.3 Creating a basis English sentiment dictionary (bESD) in a distributed system

According to [52-57], we have at least 55,000 English terms, including nouns, verbs, adjectives, etc. In this part, we calculate the valence and the polarity of the English words or phrases for our basis English sentiment dictionary (bESD) by using the YIIM in a parallel network environment, as the following diagram in Figure 22 below shows.

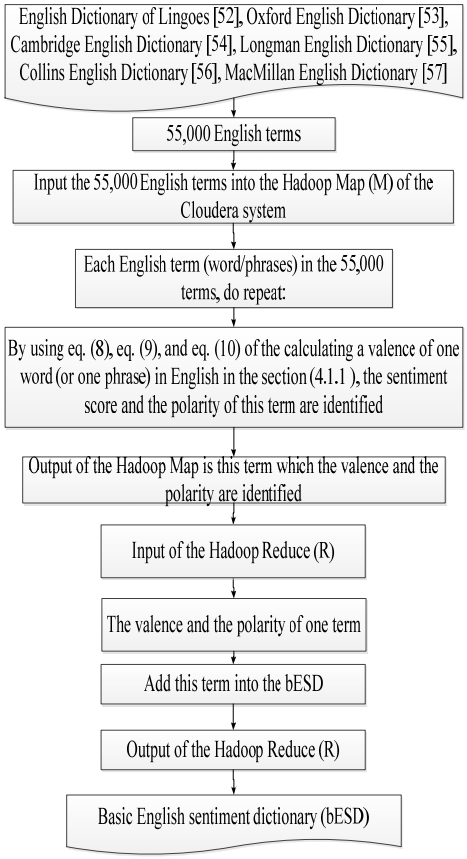

Figure 22: Overview of creating a basis English sentiment dictionary (bESD) in a distributed environment

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In Figure 22, this section includes two phases: the Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase is the 55,000 terms in English in [52-57]. The output of the Hadoop Map phase is one term which the sentiment score and the polarity are identified. The output of the Hadoop Map phase is the input of the Hadoop Reduce phase. Thus, the input of the Hadoop Reduce phase is one term which the sentiment score and the polarity are identified. The output of the Hadoop Reduce phase is the basis English sentiment dictionary (bESD).

We proposed the algorithm 26 to implement the Hadoop Map phase of creating a basis English sentiment dictionary (bESD) in a distributed environment. The main ideas of the algorithm 26 are as follows:

Input: the 55,000 English terms; the Google search engine

Output: one term which the sentiment score and the polarity are identified.

Step 1: Each term in the 55,000 terms, do repeat:

Step 2: By using eq. (8), eq. (9), and eq. (10) of the calculating a valence of one word (or one phrase) in English in the section (4.1.1), the sentiment score and the polarity of this term are identified. The valence and the polarity are calculated by using the YIIM through the Google search engine with AND operator and OR operator.

Step 3: Return this term;

We proposed the algorithm 27 to perform thie Hadoop Reduce phase of creating a basis English sentiment dictionary (bESD) in a distributed environment. The main ideas of the algorithm 27 are as follows:

Input: one term which the sentiment score and the polarity are identified – The output of the Hadoop Map phase.

Output: a basis English sentiment dictionary (bESD)

Step 1: Add this term into the basis English sentiment dictionary (bESD);

Step 2: Return bESD;

Our basis English sentiment dictionary (bESD) has more 55,000 English words (or English phrases) and bESD is stored in Microsoft SQL Server 2008 R2.

5. EXPERIMENT

We have measured Accuracy (A) to calculate the accuracy of the results of emotion classification.

We used a Java programming language for programming to save data sets, implementing our proposed model to classify the 5,500,000 documents of the testing data set and the 3,000,000 documents of the training data set. To implement the proposed model, we have already used Java programming language to save the English testing data set and to save the results of emotion classification.

The proposed model was implemented in both the sequential system and the distributed network environment.

Our model related to the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD is implemented in the sequential environment with the configuration as follows: The sequential environment in this research includes 1 node (1 server). The configuration of the server in the sequential environment is: Intel® Server Board S1200V3RPS, Intel® Pentium® Processor G3220 (3M Cache, 3.00 GHz), 2GB CC3-10600 ECC 1333 MHz LP Unbuffered DIMMs. The operating system of the server is: Cloudera. The Java language is used in programming our model related to the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD

The proposed model related to the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD is performed in the Cloudera parallel network environment with the configuration as follows: This Cloudera system includes 9 nodes (9 servers). The configuration of each server in the Cloudera system is: Intel® Server Board S1200V3RPS, Intel® Pentium® Processor G3220 (3M Cache, 3.00 GHz), 2GB CC3-10600 ECC 1333 MHz LP Unbuffered DIMMs. The operating system of each server in the 9 servers is: Cloudera. All 9 nodes have the same configuration information. The Java language is used in programming the application of the proposed model related to the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD in the Cloudera

In Table 1, the results of the documents of the English testing data set to test are presented.

The accuracy of the sentiment classification of the documents in the English testing data set is shown in Table 2 below.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The average time of the classification of our new model for the English documents in testing data set are displayed in Table 3.

6. CONCLUSION

In this survey, a new model has been proposed to classify sentiment of many documents in English using the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD with Hadoop Map (M) /Reduce (R) in the Cloudera parallel network environment. Based on our proposed new model, we have achieved 88.72% accuracy of the testing data set in Table 2. Until now, not many studies have shown that the clustering methods can be used to classify data. Our research shows that clustering methods are used to classify data and, in particular, can be used to classify the sentiments (positive, negative, or neutral) in text.

The proposed model can be applied to other languages although our new model has been tested on our English data set. Our model can be applied to larger data sets with millions of English documents in the shortest time although our model has been tested on the documents of the testing data set in which the data sets are small in this survey.

According to Table 3, the average time of the sentiment classification of using the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD in the sequential environment is 22,508,676 seconds / 5,500,000 English documents and it is greater than the average time of the sentiment classification of using the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the the sentiment lexicons of the bESD in the Cloudera parallel network environment with 3 nodes which is 8,069,558 seconds / 5,500,000 English documents. The average time of the sentiment classification of using the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD in the Cloudera parallel network environment with 9 nodes is 2,623,186 seconds / 5,500,000 English documents, and It is the shortest time in the table. Besides, The average time of the sentiment classification of using the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD in the Cloudera parallel network environment with 6 nodes is 3,984,779 seconds / 5,500,000 English documents

The execution time of using the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD in the Cloudera is dependent on the performance of the Cloudera parallel system and also dependent on the performance of each server on the Cloudera system.

The accuracy of the proposed model is depending on many factors as follows:

1)The SOM – related algorithms
2)The testing data set
3)The documents of the testing data set must be standardized carefully.
4)Transferring one document into one multi-dimensional vector

The execution time of the proposed model is depending on many factors as follows:

1)The parallel network environment such as the Cloudera system.
2)The distributed functions such as Hadoop Map (M) and Hadoop Reduce (R).
3)The SOM – related algorithms
4)The performance of the distributed network system.
5)The number of nodes of the parallel network environment.
6)The performance of each node (each server) of the distributed environment.
7)The sizes of the training data set and the testing data set.
8)Transferring one document into one one-dimensional vector.

The proposed model has many advantages and disadvantages. Its positives are as follows: It uses the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons to classify semantics of English documents based on sentences. The proposed model can process millions of documents in the shortest time. This study can be performed in distributed systems to shorten the execution time of the proposed model. It can be applied to other languages. Its negatives are as follows: It has a low rate of accuracy. It costs too much and takes too much time to implement this proposed model.

To understand the scientific values of this research, we have compared our model's results with many studies in the tables below.

In Table 4, we present the comparisons of our model's results with the works in [1-3]

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The comparisons of our model's advantages and disadvantages with the works in [1-3] are shown in Table 5.

In Table 6, we display the comparisons of our model with the latest sentiment classification models (or the latest sentiment classification methods) in [4-14]

The comparisons of our model's positives and negatives with the latest sentiment classification models (or the latest sentiment classification methods) in [4-14] are presented in Table 7.

In Table 8, we show the comparisons of our model with the researches related to Self-Organizing Map Algorithm (SOM) in [15-19]

The comparisons of our model's positives and negatives with the surveys related to the Self-Organizing Map Algorithm (SOM) in [15-19] are displayed in Table 9.

Future Work

Based on the results of this proposed model, many future projects can be proposed, such as creating full emotional lexicons in a parallel network environment to shorten execution times, creating many search engines, creating many translation engines, creating many applications that can check grammar correctly. This model can be applied to many different languages, creating applications that can analyze the emotions of texts and speeches, and machines that can analyze sentiments.

REFRENCES:

[1] Vaibhav Kant Singh, Vinay Kumar Singh, "*Vector Space Model: An Information Retrieval System*", Int. J. Adv. Engg. Res. Studies/IV/II/Jan.-March,2015/141-143, 2015

[2] Víctor Carrera-Trejo, Grigori Sidorov, Sabino Miranda-Jiménez, Marco Moreno Ibarra and Rodrigo Cadena Martínez, "*Latent Dirichlet Allocation complement in the vector space model for Multi-Label Text Classification*", International Journal of Combinatorial Optimization Problems and Informatics, Vol. 6, No. 1, 2015, pp. 7-19.

[3] Pascal Soucy, Guy W. Mineau, "*Beyond TFIDF Weighting for Text Categorization in the Vector Space Model*", Proceedings of the 19th International Joint Conference on Artificial Intelligence, 2015, pp. 1130-1135, USA.

[4] Basant Agarwal, Namita Mittal, "*Machine Learning Approach for Sentiment Analysis*", Prominent Feature Extraction for Sentiment Analysis, Print ISBN 978-3-319-25341-1, 10.1007/978-3-319-25343-5_3, 2016, 21-45.

[5] Basant Agarwal, Namita Mittal, "*Semantic Orientation-Based Approach for Sentiment Analysis*", Prominent Feature Extraction for Sentiment Analysis, Print ISBN 978-3-319-25341-1, 10.1007/978-3-319-25343-5_6, 2016, 77-88

[6] Sérgio Canuto, Marcos André, Gonçalves, Fabrício Benevenuto, "*Exploiting New Sentiment-Based Meta-level Features for Effective Sentiment Analysis*", Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM '16), 2016, 53-62, New York USA.

[7] Shoiab Ahmed, Ajit Danti, "*Effective Sentimental Analysis and Opinion Mining of Web Reviews Using Rule Based Classifiers",* Computational Intelligence in Data Mining, Volume 1, Print ISBN 978-81-322-2732-8, DOI 10.1007/978-81-322-2734-2_18, 2016, 171-179, India.

[8] Vo Ngoc Phu, Phan Thi Tuoi, "*Sentiment classification using Enhanced Contextual Valence Shifters*", International Conference on Asian Language Processing (IALP), 2014, 224-229.

[9] Vo Thi Ngoc Tran, Vo Ngoc Phu and Phan Thi Tuoi, "*Learning More Chi Square Feature Selection to Improve the Fastest and Most Accurate Sentiment Classification*", The Third Asian Conference on Information Systems (ACIS 2014), 2014

[10] Nguyen Duy Dat, Vo Ngoc Phu, Vo Thi Ngoc Tran, Vo Thi Ngoc Chau, "*STING Algorithm used English Sentiment Classification in A Parallel Environment"*, International Journal of Pattern Recognition and Artificial Intelligence, January 2017.

[11] Vo Ngoc Phu, Nguyen Duy Dat, Vo Thi Ngoc Tran, Vo Thi Ngoc Tran, "*Fuzzy C-Means for English Sentiment Classification in a Distributed System*", International Journal of Applied Intelligence (APIN), DOI: 10.1007/s10489-016-0858-z, November 2016, 1-22.

[12] Vo Ngoc Phu, Chau Vo Thi Ngoc, Tran Vo THi Ngoc, Dat Nguyen Duy, "*A C4.5 algorithm for english emotional classification"*, Evolving Systems, pp 1-27, doi:10.1007/s12530-017-9180-1, April 2017.

[13] Vo Ngoc Phu, Vo Thi Ngoc Chau, Vo Thi Ngoc Tran, "*SVM for English Semantic Classification in Parallel Environment*", International Journal

of Speech Technology (IJST), 10.1007/s10772-017-9421-5, May 2017, 31 pages.

[14] Vo Ngoc Phu, Vo Thi Ngoc Tran, Vo Thi Ngoc Chau, Nguyen Duy Dat, Khanh Ly Doan Duy, "*A Decision Tree using ID3 Algorithm for English Semantic Analysis*", International Journal of Speech Technology (IJST), DOI: 10.1007/s10772-017-9429-x, 2017, 23 pages

[15] T. Kohonen, "*The self-organizing map*", Proceedings of the IEEE, Volume: 78, Issue: 9,DOI: 10.1109/5.58325, 1990

[16] J.L. Giraudel, S. Lek, "*A comparison of self-organizing map algorithm and some conventional statistical methods for ecological community ordination*", Ecological Modelling, Volume 146, Issues 1–3, https://doi.org/10.1016/S0304-3800(01)00324-6, 2001, Pages 329-339

[17] J. Vesanto; E. Alhoniemi, "*Clustering of the self-organizing map*", IEEE Transactions on Neural Networks, Volume: 11, Issue: 3,DOI: 10.1109/72.846731, 2000

[18] Roussinov, Dmitri G.; Chen, Hsinchun, "*A Scalable Self-organizing Map Algorithm for Textual Classification: A Neural Network Approach to Thesaurus Generation*", Communication and Cognition in Artificial Intelligence Journal, 15(1-2):81-111, 1998

[19] E. Berglund; J. Sitte, "*The parameterless self-organizing map algorithm*", IEEE Transactions on Neural Networks, Volume: 17, Issue: 2, DOI: 10.1109/TNN.2006.871720, 2006

[20] Aleksander Bai, Hugo Hammer, "*Constructing sentiment lexicons in Norwegian from a large text corpus*", 2014 IEEE 17th International Conference on Computational Science and Engineering, 2014

[21] P.D.Turney, M.L.Littman, "*Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus*", arXiv:cs/0212012, Learning (cs.LG); Information Retrieval (cs.IR), 2002

22. Robert Malouf, Tony Mullen (2017) Graph-based user classification for informal online political discourse. In proceedings of the 1st Workshop on Information Credibility on the Web.

[23] Christian Scheible, "*Sentiment Translation through Lexicon Induction*", Proceedings of the ACL 2010 Student Research Workshop, Sweden, 2010, pp 25–30.

[24] Dame Jovanoski, Veno Pachovski, Preslav Nakov, "*Sentiment Analysis in Twitter for Macedonian*", Proceedings of Recent Advances in Natural Language Processing, Bulgaria, 2015, pp 249–257.

[25] Amal Htait, Sebastien Fournier, Patrice Bellot, "*LSIS at SemEval-2016 Task 7: Using Web Search Engines for English and Arabic Unsupervised Sentiment Intensity Prediction*", Proceedings of SemEval-2016, California, 2016, pp 481–485.

[26] Xiaojun Wan, "*Co-Training for Cross-Lingual Sentiment Classification*", Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, Singapore, 2009, pp 235–243.

[27] Julian Brooke, Milan Tofiloski, Maite Taboada, "*Cross-Linguistic Sentiment Analysis: From English to Spanish*", International Conference RANLP 2009 - Borovets, Bulgaria, 2009, pp 50–54.

[28] Tao Jiang, Jing Jiang, Yugang Dai, Ailing Li, "*Micro–blog Emotion Orientation Analysis Algorithm Based on Tibetan and Chinese Mixed Text*", International Symposium on Social Science (ISSS 2015), 2015

[29] Tan, S.; Zhang, J., "*An empirical study of sentiment analysis for Chinese documents*", Expert Systems with Applications, doi:10.1016/j.eswa.2007.05.028, 2017

[30] Weifu Du, Songbo Tan, Xueqi Cheng, Xiaochun Yun, "*Adapting Information Bottleneck Method for Automatic Construction of Domain-oriented Sentiment Lexicon*", WSDM'10, New York, USA, 2010

[31] Ziqing Zhang, Qiang Ye, Wenying Zheng, Yijun Li, "*Sentiment Classification for Consumer Word-of-Mouth in Chinese: Comparison between Supervised and Unsupervised Approaches*", The 2010 International Conference on E-Business Intelligence, 2010

[32] Guangwei Wang, Kenji Araki, "*Modifying SO-PMI for Japanese Weblog Opinion Mining by Using a Balancing Factor and Detecting Neutral Expressions*", Proceedings of NAACL HLT 2007, Companion Volume, NY, 2007, pp 189–192.

[33] Shi Feng, Le Zhang, Binyang Li Daling Wang, Ge Yu, Kam-Fai Wong, "Is Twitter A Better Corpus for Measuring Sentiment Similarity?", Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, USA, 2013, pp 897–902.

[34] Nguyen Thi Thu An, Masafumi Hagiwara, "*Adjective-Based Estimation of Short*

Sentence's Impression", (KEER2014) Proceedings of the 5th Kanesi Engineering and Emotion Research; International Conference; Sweden, 2014.

[35] Nihalahmad R. Shikalgar, Arati M. Dixit, "*JIBCA: JaYIIMard Index based Clustering Algorithm for Mining Online Review*", International Journal of Computer Applications (0975 – 8887), Volume 105 – No. 15, 2014.

[36] Xiang Ji, Soon Ae Chun, Zhi Wei, James Geller, "*Twitter sentiment classification for measuring public health concerns*", Soc. Netw. Anal. Min. (2015) 5:13, DOI 10.1007/s13278-015-0253-5, 2015

[37] Nazlia Omar, Mohammed Albared, Adel Qasem Al-Shabi, Tareg Al-Moslmi, "*Ensemble of Classification algorithms for Subjectivity and Sentiment Analysis of Arabic Customers' Reviews*", International Journal of Advancements in Computing Technology(IJACT), Volume 5, 2013

[38] Huina Mao, Pengjie Gao, Yongxiang Wang, Johan Bollen, "*Automatic Construction of Financial Semantic Orientation Lexicon from Large-Scale Chinese News Corpus*", 7th Financial Risks International Forum, Institut Louis Bachelier, 2014

[39] Yong REN, Nobuhiro KAJI, Naoki YOSHINAGA, Masaru KITSUREGAW, "*Sentiment Classification in Under-Resourced Languages Using Graph-based Semi-supervised Learning Methods*", IEICE TRANS. INF. & SYST., VOL.E97–D, NO.4, DOI: 10.1587/transinf.E97.D.1, 2014

[40] Oded Netzer, Ronen Feldman, Jacob Goldenberg, Moshe Fresko, "*Mine Your Own Business: Market-Structure Surveillance Through Text Mining*", Marketing Science, Vol. 31, No. 3, 2012, pp 521-543.

[41] Yong Ren, Nobuhiro Kaji, Naoki Yoshinaga, Masashi Toyoda, Masaru Kitsuregawa, "*Sentiment Classification in Resource-Scarce Languages by using Label Propagation*", Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation, Institute of Digital Enhancement of Cognitive Processing, Waseda University, 2011, pp 420 - 429.

[42] José Alfredo Hernández-Ugalde, Jorge Mora-Urpí, Oscar J. Rocha, "*Genetic relationships among wild and cultivated populations of peach palm (Bactris gasipaes Kunth, Palmae): evidence for multiple independent domestication events*", Genetic Resources and Crop Evolution, Volume 58, Issue 4, 2011, pp 571-583.

[43] Julia V. Ponomarenko, Philip E. Bourne, Ilya N. Shindyalov, "*Building an automated classification of DNA-binding protein domains*", BIOINFORMATICS, Vol. 18, 2002, pp S192-S201.

[44] Andréia da Silva Meyer, Antonio Augusto Franco Garcia, Anete Pereira de Souza, Cláudio Lopes de Souza Jr, "*Comparison of similarity coefficients used for cluster analysis with dominant markers in maize (Zea maysL)*", Genetics and Molecular Biology, 27, 1, 2004, 83-91.

[45] Snežana MLADENOVIĆ DRINIĆ, Ana NIKOLIĆ, Vesna PERIĆ, "*Cluster Analysis of Soybean Genotypes Based on RAPD Markers*", Proceedings 43rd Croatian and 3rd International Symposium on Agriculture. Opatija. Croatia, 2008, 367- 370.

[46] Tamás, Júlia; Podani, János; Csontos, Péter, "*An extension of presence/absence coefficients to abundance data:a new look at absence*", Journal of Vegetation Science 12: 401-410, 2001.

[47] Vo Ngoc Phu, Vo Thi Ngoc Chau, Vo Thi Ngoc Tran, Nguyen Duy Dat, "*A Vietnamese adjective emotion dictionary based on exploitation of Vietnamese language characteristics*", International Journal of Artificial Intelligence Review (AIR), doi:10.1007/s10462-017-9538-6, 2017, 67 pages.

[48] Vo Ngoc Phu, Vo Thi Ngoc Chau, Nguyen Duy Dat, Vo Thi Ngoc Tran, Tuan A. Nguyen, "*A Valences-Totaling Model for English Sentiment Classification*", International Journal of Knowledge and Information Systems, DOI: 10.1007/s10115-017-1054-0, 2017, 30 pages.

[49] Vo Ngoc Phu, Vo Thi Ngoc Chau, Vo Thi Ngoc Tran, "*Shifting Semantic Values of English Phrases for Classification*", International Journal of Speech Technology (IJST), 10.1007/s10772-017-9420-6, 2017, 28 pages.

[50] Vo Ngoc Phu, Vo Thi Ngoc Chau, Vo Thi Ngoc Tran, Nguy Duy Dat, Khanh Ly Doan Duy, "*A Valence-Totaling Model for Vietnamese Sentiment Classification*", International Journal of Evolving Systems (EVOS), DOI: 10.1007/s12530-017-9187-7, 2017, 47 pages.

[51] Vo Ngoc Phu, Vo Thi Ngoc Chau, Vo Thi Ngoc Tran, Nguyen Duy Dat, Khanh Ly Doan Duy, "*Semantic Lexicons of English Nouns for Classification*", International Journal of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Evolving Systems, DOI: 10.1007/s12530-017-9188-6, 2017, 69 pages.

[52] English Dictionary of Lingoes, http://www.lingoes.net/,2017

[53] Oxford English Dictionary, http://www.oxforddictionaries.com/,2017

[54] Cambridge English Dictionary, http://dictionary.cambridge.org/, 2017

[55] Longman English Dictionary, http://www.ldoceonline.com/, 2017

[56] Collins English Dictionary, http://www.collinsdictionary.com/dictionary/english,2017

[57] MacMillan English Dictionary, http://www.macmillandictionary.com/,2017

[58] Seung-Seok Choi, Sung-Hyuk Cha, Charles C. Tappert, "*A Survey Of Binary Similarity And Distance Measures*", Systemics, Cybernetics And Informatics, Issn: 1690-4524, Volume 8 - Number 1

[59] Yadolah Dodge, "*Yule and Kendall Coefficient*", The Concise Encyclopedia of Statistics, DOI10.1007/978-0-387-32833-1_431, 2008, pp 581-583

[60] Matthijs J. Warrens, "*On Association Coefficients for 2×2 Tables and Properties That Do Not Depend on the Marginal Distributions*", Psychometrika. 2008 Dec; 73(4): 777–789, doi: 10.1007/s11336-008-9070-3, 2008

[61] Douglas G. Bonett, Robert M. Price, "*Statistical Inference for Generalized Yule Coefficients in 2 × 2 Contingency Tables*", Sociological Methods & Research, Vol 35, issue 3, 2007

[62] David P. Doane, Lori E. Seward, "*Measuring Skewness: A Forgotten Statistic?*", Journal of Statistics Education Volume 19, Number 2, 2011

[63] Donald A. Jackson, Keith M. Somers, Harold H. Harvey, "*Similarity Coefficients: Measures of Co-occurrence and Association or Simply Measures of Occurrence?"* The American Naturalist, Vol 133, No 3, DOI: 10.1086/284927, 1989

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

APPENDICES:

Table 1: The results of the English documents in the testing data set.

Table 2: The accuracy of our new model for the English documents in the testing data set.

Table 3: Average time of the classification of our new model for the English documents in testing data set.

Table 4: Comparisons of our model's results with the works in [1-3]

Table 5: Comparisons of our model's advantages and disadvantages with the works in [1-3]

Table 6: Comparisons of our model with the latest sentiment classification models (or the latest sentiment classification methods) in [4-14]

Table 7: Comparisons of our model's positives and negatives with the latest sentiment classification models (or the latest sentiment classification methods) in [4-14]

Table 8: Comparisons of our model with the researches related to Self-Organizing Map Algorithm (SOM) in [15-19]

Table 9: Comparisons of our model's positives and negatives the surveys related to the Self-Organizing Map Algorithm (SOM) in [15-19]

Table 10: Comparisons of our model's results with the works related to [20-51].

Table 11: Comparisons of our model's advantages and disadvantages with the works related to [20-51].

Table 12: Comparisons of our model's results with the works related to the YULE-II coefficient (MC) in [58-63]

Table 13: Comparisons of our model's benefits and drawbacks with the studies related to the YULE-II coefficient (MC) in [58-63]

Table 1: The results of the English documents in the testing data set.

	Testing Dataset	**Correct Classification**	**Incorrect Classification**
Negative	2,750,000	2,440,722	309,278
Positive	2,750,000	2,438,878	311,122
Summary	5,500,000	4,879,600	620,400

Table 2: The accuracy of our new model for the English documents in the testing data set.

Proposed Model	Class	Accuracy
Our new model	Negative	88.72 %
	Positive	

Table 3: Average time of the classification of our new model for the English documents in testing data set.

	Average time of the classification / 5,500,000 English documents.
The Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD in the sequential environment	22,508,676 seconds
The Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD in the Cloudera distributed system with 3 nodes	8,069,558 seconds

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	Average time of the classification / 5,500,000 English documents.
The Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD in the Cloudera distributed system with 6 nodes	3,984,779 seconds
The Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD in the Cloudera distributed system with 9 nodes	2,623,186 seconds

Table 4: Comparisons of our model's results with the works in [1-3]

Clustering technique: CT.
Parallel network system: PNS (distributed system).
Special Domain: SD.
Depending on the training data set: DT.
Vector Space Model: VSM
No Mention: NM
English Language: EL.

Studies	**CT**	**Sentiment Classification**	**PNS**	**SD**	**DT**	**Language**	**VSM**
[1]	No	No	No	Yes	No	EL	Yes
[2]	No	Yes	No	Yes	No	EL	Yes
[3]	No	Yes	No	Yes	Yes	EL	Yes
Our work	Yes	Yes	Yes	Yes	Yes	EL	Yes

Table 5: Comparisons of our model's advantages and disadvantages with the works in [1-3]

Researches	Approach	Advantages	Disadvantages
[1]	Examining the vector space model, an information retrieval technique and its variation	In this work, the authors have given an insider to the working of vector space model techniques used for efficient retrieval techniques. It is the bare fact that each system has its own strengths and weaknesses. What we have sorted out in the authors' work for vector space modeling is that the model is easy to understand and cheaper to implement, considering the fact that the system should be cost effective (i.e., should follow the space/time constraint. It is also very popular. Although the system has all these properties, it is facing some major drawbacks.	The drawbacks are that the system yields no theoretical findings. Weights associated with the vectors are very arbitrary, and this system is an independent system, thus requiring separate attention. Though it is a promising technique, the current level of success of the vector space model techniques used for information retrieval are not able to satisfy user needs and need extensive attention.
[2]	+Latent Dirichlet allocation (LDA). +Multi-label text classification tasks and apply various feature sets. +Several combinations of features, like bi-grams and uni-grams.	In this work, the authors consider multi-label text classification tasks and apply various feature sets. The authors consider a subset of multi-labeled files of the Reuters-21578 corpus. The authors use traditional TF-IDF values of the features and tried both considering and ignoring stop words. The authors also tried several combinations of features, like bi-grams and uni-grams. The authors also experimented with adding LDA results into vector space models as new features. These last experiments obtained the best results.	No mention

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[3]	The K-Nearest Neighbors algorithm for English sentiment classification in the Cloudera distributed system.	In this study, the authors introduce a new weighting method based on statistical estimation of the importance of a word for a specific categorization problem. One benefit of this method is that it can make feature selection implicit, since useless features of the categorization problem considered get a very small weight. Extensive experiments reported in the work show that this new weighting method improves significantly the classification accuracy as measured on many categorization tasks.	Despite positive results in some settings, GainRatio failed to show that supervised weighting methods are generally higher than unsupervised ones. The authors believe that ConfWeight is a promising supervised weighting technique that behaves gracefully both with and without feature selection. Therefore, the authors advocate its use in further experiments.
Our work		-We use the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. The advantages and disadvantages of the proposed model are shown in the Conclusion section.	

Table 6: Comparisons of our model with the latest sentiment classification models (or the latest sentiment classification methods) in [4-14]

Studies	CT	Sentiment Classification	PNS	SD	DT	Language	VSM
[4]	No	Yes	NM	Yes	Yes	Yes	vector
[5]	No	Yes	NM	Yes	Yes	NM	NM
[6]	No	Yes	NM	Yes	Yes	EL	NM
[7]	No	Yes	NM	Yes	Yes	NM	NM
[8]	No	Yes	No	No	No	EL	No
[9]	No	Yes	No	No	No	EL	No
Our work	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 7: Comparisons of our model's positives and negatives the latest sentiment classification models (or the latest sentiment classification methods) in [4-14]

Studies	Approach	Positives	Negatives
[4]	The Machine Learning Approaches Applied to Sentiment Analysis-Based Applications	The main emphasis of this survey is to discuss the research involved in applying machine learning methods, mostly for sentiment classification at document level. Machine learning-based approaches work in the following phases, which are discussed in detail in this work for sentiment classification: (1) feature extraction, (2) feature weighting schemes, (3) feature selection, and (4) machine-learning methods. This study also discusses the standard free benchmark datasets and evaluation methods for sentiment analysis. The authors conclude the research with a comparative study of some state-of-the-art methods for sentiment analysis and some possible future research directions in opinion mining and sentiment analysis.	No mention
[5]	Semantic Orientation-Based Approach for Sentiment Analysis	This approach initially mines sentiment-bearing terms from the unstructured text and further computes the polarity of the terms. Most of the sentiment-bearing terms are multi-word features unlike bag-of-words, e.g., "good movie," "nice cinematography," "nice actors," etc. Performance of semantic orientation-based approach has been limited in the literature due to inadequate coverage of multi-word features.	No mention
[6]	Exploiting New Sentiment-Based Meta-Level Features for Effective Sentiment Analysis	Experiments performed with a substantial number of datasets (nineteen) demonstrate that the effectiveness of the proposed sentiment-based meta-level features is not only superior to the traditional bag-of-words representation (by up to 16%) but also is also superior in most cases to state-of-art meta-level features previously proposed in the literature for text classification tasks that do not take into account any	A line of future research would be to explore the authors'

		idiosyncrasies of sentiment analysis. The authors' proposal is also largely superior to the best lexicon-based methods as well as to supervised combinations of them. In fact, the proposed approach is the only one to produce the best results in all tested datasets in all scenarios.	meta features with other classificati on algorithms and feature selection techniques in different sentiment analysis tasks such as scoring movies or products according to their related reviews.
[7]	Rule-Based Machine Learning Algorithms	The proposed approach is tested by experimenting with online books and political reviews and demonstrates the efficacy through Kappa measures, which have a higher accuracy of 97.4% and a lower error rate. The weighted average of different accuracy measures like Precision, Recall, and TP-Rate depicts higher efficiency rate and lower FP-Rate. Comparative experiments on various rule-based machine learning algorithms have been performed through a ten-fold cross validation training model for sentiment classification.	No mention
[8]	The Combination of Term-Counting Method and Enhanced Contextual Valence Shifters Method	The authors have explored different methods of improving the accuracy of sentiment classification. The sentiment orientation of a document can be positive (+), negative (-), or neutral (0). The authors combine five dictionaries into a new one with 21,137 entries. The new dictionary has many verbs, adverbs, phrases and idioms that were not in five dictionaries before. The study shows that the authors' proposed method based on the combination of Term-Counting method and Enhanced Contextual Valence Shifters method has improved the accuracy of sentiment classification. The combined method has accuracy 68.984% on the testing dataset, and 69.224% on the training dataset. All of these methods are implemented to classify the reviews based on our new dictionary and the Internet Movie Data based data set.	No mention
[9]	Naive Bayes Model with N-GRAM Method, Negation Handling Method, Chi-Square Method and Good-Turing Discounting, etc.	The authors have explored the Naive Bayes model with N-GRAM method, Negation Handling method, Chi-Square method and Good-Turing Discounting by selecting different thresholds of Good-Turing Discounting method and different minimum frequencies of Chi-Square method to improve the accuracy of sentiment classification.	No Mention
Our work	-We use the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. The positives and negatives of the proposed model are given in the Conclusion section.		

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 8: Comparisons of our model with the researches related to Self-Organizing Map Algorithm (SOM) in [15-19]

Studies	CT	Sentiment Classification	PNS	SD	DT	Language	VSM
[15]	Yes	No	NM	No	No	Yes	vector
[16]	Yes	No	NM	No	No	NM	NM
[17]	Yes	No	NM	No	No	EL	NM
[18]	Yes	No	NM	No	No	NM	NM
[19]	Yes	No	No	No	No	EL	No
Our work	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 9: Comparisons of our model's positives and negatives the surveys related to the Self-Organizing Map Algorithm (SOM) in [15-19]

Studies	Approach	Positives	Negatives
[15]	The self-organizing map	One result of this is that the self-organization process can discover semantic relationships in sentences. Brain maps, semantic maps, and early work on competitive learning are reviewed. The self-organizing map algorithm (an algorithm which order responses spatially) is reviewed, focusing on best matching cell selection and adaptation of the weight vectors. Suggestions for applying the self-organizing map algorithm, demonstrations of the ordering process, and an example of hierarchical clustering of data are presented. Fine tuning the map by learning vector quantization is addressed. The use of self-organized maps in practical speech recognition and a simulation experiment on semantic mapping are discussed.	No mention
[16]	A comparison of self-organizing map algorithm and some conventional statistical methods for ecological community ordination	After the presentation of SOM adapted to ecological data, SOM was trained on popular example data; upland forest in Wisconsin (USA). The SOM results were compared with classical statistical techniques. Similarity between the results may be observed and constitutes a validation of the SOM method. SOM algorithm seems fully usable in ecology, it can perfectly complete classical techniques for exploring data and for achieving community ordination.	No mention
[17]	Clustering of the self-organizing map	In this study, different approaches to clustering of the SOM are considered. In particular, the use of hierarchical agglomerative clustering and partitive clustering using K-means are investigated. The two-stage procedure-first using SOM to produce the prototypes that are then clustered in the second stage-is found to perform well when compared with direct clustering of the data and to reduce the computation time.	No mention
[18]	A Scalable Self-organizing Map Algorithm for Textual Classification: A Neural Network Approach to Thesaurus Generation	The authors' proposed data structure and algorithm took advantage of the sparsity of coordinates in the document input vectors and reduced the SOM computational complexity by several order of magnitude. The proposed Scaleable SOM (SSOM) algorithm makes large-scale textual categorization tasks a possibility. Algorithmic intuition and the mathematical foundation of the authors' research are presented in detail. The authors also describe three benchmarking experiments to examine the algorithm's performance at various scales: classification of electronic meeting comments, Internet homepages, and the Compendex collection.	No mention
[19]	The parameterless self-organizing map algorithm	The authors discuss the relative performance of the PLSOM and the SOM and demonstrate some tasks in which the SOM fails but the PLSOM performs satisfactory. Finally the authors discuss some example applications of the PLSOM and present a proof of ordering under certain limited conditions.	No mention

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Our work	-We use the Self-Organizing Map algorithm, a testing data the multi-dimensional vectors set and a training data set with based on the sentiment lexicons of the bESD to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. The positives and negatives of the proposed model are given in the Conclusion section.

Table 10: Comparisons of our model's results with the works related to [20-51].

YULE-II MEASURE coefficient (YIIM)

Semantic classification, sentiment classification: SC

Studies	PMI	JM	Language	SD	DT	YIIM	SC	Other measures	Search engines
[20]	Yes	No	English	Yes	Yes	No	Yes	No	No Mention
[21]	Yes	No	English	Yes	No	No	Yes	Latent Semantic Analysis (LSA)	AltaVista
[22]	Yes	No	English	Yes	Yes	No	Yes	Baseline; Turney-inspired; NB; Cluster+NB; Human	AltaVista
[23]	Yes	No	English German	Yes	Yes	No	Yes	SimRank	Google search engine
[24]	Yes	No	English Macedonian	Yes	Yes	No	Yes	No Mention	AltaVista search engine
[25]	Yes	No	English Arabic	Yes	No	No	Yes	No Mention	Google search engine Bing search engine
[26]	Yes	No	English Chinese	Yes	Yes	No	Yes	SVM(CN); SVM(EN); SVM(ENCN1); SVM(ENCN2); TSVM(CN); TSVM(EN); TSVM(ENCN1); TSVM(ENCN2); CoTrain	No Mention
[27]	Yes	No	English Spanish	Yes	Yes	No	Yes	SO Calculation SVM	Google
[28]	Yes	No	Chinese Tibetan	Yes	Yes	No	Yes	- Feature selection -Expectation Cross Entropy -Information Gain	No Mention
[29]	Yes	No	Chinese	Yes	Yes	No	Yes	DF, CHI, MI andIG	No Mention
[30]	Yes	No	Chinese	Yes	No	No	Yes	Information Bottleneck Method (IB); LE	AltaVista
[31]	Yes	No	Chinese	Yes	Yes	No	Yes	SVM	Google Yahoo Baidu
[32]	Yes	No	Japanese	No	No	No	Yes	Harmonic−Mean	Google and replaced the NEAR operator with the AND operator inthe SO

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

									formula.
[33]	Yes	Yes	English	Yes	Yes	No	Yes	Dice; NGD	Google search engine
[34]	Yes	Yes	English	Yes	No	No	Yes	Dice; Overlap	Google
[35]	No	Yes	English	Yes	Yes	No	Yes	A JaYIIMard index based clustering algorithm (JIBCA)	No Mention
[36]	No	Yes	English	Yes	Yes	No	Yes	Naıve Bayes, Two-Step Multinomial Naıve Bayes, and Two-Step Polynomial-Kernel Support Vector Machine	Google
[37]	No	Yes	Arabic	No	No	No	Yes	Naive Bayes (NB); Support Vector Machines (SVM); RoYIIMhio; Cosine	No Mention
[38]	No	Yes	Chinese	Yes	Yes	No	Yes	A new score–Economic Value (EV), etc.	Chinese search
[39]	No	Yes	Chinese	Yes	Yes	No	Yes	Cosine	No Mention
[40]	No	Yes	English	No	Yes	No	Yes	Cosine	No Mention
[41]	No	Yes	Chinese	No	Yes	No	Yes	Dice; overlap; Cosine	No Mention
[42]	No	No	Vietnam ese	No	No	No	Yes	Ochiai Measure	Google
[43]	No	No	English	No	No	No	Yes	Cosine coefficient	Google
[44]	No	No	English	No	No	No	Yes	Sorensen measure	Google
[45]	No	Yes	Vietnam ese	No	No	No	Yes	JaYIIMard	Google
[46]	No	No	English	No	No	No	Yes	Tanimoto coefficient	Google
Our work	No	No	English Languag e	No	No	Yes	Yes	No	Google search engine

Table 11: Comparisons of our model's advantages and disadvantages with the works related to [20-51].

Surveys	Approach	Advantages	Disadvantages
[20]	Constructing sentiment lexicons in Norwegian from a large text corpus	Through the authors' PMI computations in this survey they used a distance of 100 words from the seed word, but it might be that other lengths that generate better sentiment lexicons. Some of the authors' preliminary research showed that 100 gave a better result.	The authors need to investigate this more closely to find the optimal distance. Another factor that has not been investigated much in the literature is the selection of seed words. Since they are the basis for PMI calculation, it might be a lot to gain by finding better seed words. The authors would like to explore the impact that different approaches to seed word selection have on the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

			performance of the developed sentiment lexicons.
[21]	Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus.	This survey has presented a general strategy for learning semantic orientation from semantic association, SO-A. Two instances of this strategy have been empirically evaluated, SO-PMI-IR and SO-LSA. The Accuracy of SO-PMI-IR is comparable to the Accuracy of HM, the algorithm of Hatzivassiloglou and McKeown (1997). SO-PMI-IR requires a large corpus, but it is simple, easy to implement, unsupervised, and it is not restricted to adjectives.	No Mention
[22]	Graph-based user classification for informal online political discourse	The authors describe several experiments in identifying the political orientation of posters in an informal environment. The authors' results indicate that the most promising approach is to augment text classification methods by exploiting information about how posters interact with each other	There is still much left to investigate in terms of optimizing the linguistic analysis, beginning with spelling correction and working up to shallow parsing and co-reference identification. Likewise, it will also be worthwhile to further investigate exploiting sentiment values of phrases and clauses, taking cues from methods
[23]	A novel, graph-based approach using SimRank.	The authors presented a novel approach to the translation of sentiment information that outperforms SOPMI, an established method. In particular, the authors could show that SimRank outperforms SO-PMI for values of the threshold x in an interval that most likely leads to the correct separation of positive, neutral, and negative adjectives.	The authors' future work will include a further examination of the merits of its application for knowledge-sparse languages.
[24]	Analysis in Twitter for Macedonian	The authors' experimental results show an F1-score of 92.16, which is very strong and is on par with the best results for English, which were achieved in recent SemEval competitions.	In future work, the authors are interested in studying the impact of the raw corpus size, e.g., the authors could only collect half a million tweets for creating lexicons and analyzing/evaluating the system, while Kiritchenko et al. (2014) built their lexicon on million tweets and evaluated their system on 135 million English tweets. Moreover, the authors are interested not only in quantity but also in quality, i.e., in studying the quality of the individual words and phrases

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

			used as seeds.
[25]	Using Web Search Engines for English and Arabic Unsupervised Sentiment Intensity Prediction	- For the General English sub-task, the authors' system has modest but interesting results. - For the Mixed Polarity English sub-task, the authors' system results achieve the second place. - For the Arabic phrases sub-task, the authors' system has very interesting results since they applied the unsupervised method only	Although the results are encouraging, further investigation is required, in both languages, concerning the choice of positive and negative words which once associated to a phrase, they make it more negative or more positive.
[26]	Co-Training for Cross-Lingual Sentiment Classification	The authors propose a co-training approach to making use of unlabeled Chinese data. Experimental results show the effectiveness of the proposed approach, which can outperform the standard inductive classifiers and the transductive classifiers.	In future work, the authors will improve the sentiment classification Accuracy in the following two ways: 1) The smoothed co-training approach used in (Mihalcea, 2004) will be adopted for sentiment classification. 2) The authors will employ the structural correspondence learning (SCL) domain adaption algorithm used in (Blitzer et al., 2007) for linking the translated text and the natural text.
[27]	Cross-Linguistic Sentiment Analysis: From English to Spanish	Our Spanish SO calculator (SOCAL) is clearly inferior to the authors' English SO-CAL, probably the result of a number of factors, including a small, preliminary dictionary, and a need for additional adaptation to a new language. Translating our English dictionary also seems to result in significant semantic loss, at least for original Spanish texts.	No Mention
[28]	Micro–blog Emotion Orientation Analysis Algorithm Based on Tibetan and Chinese Mixed Text	By emotion orientation analyzing and studying of Tibetan microblog which is concerned in Sina, making Tibetan Chinese emotion dictionary, Chinese sentences, Tibetan part of speech sequence and emotion symbol as emotion factors and using expected cross entropy combined fuzzy set to do feature selection to realize a kind of microblog emotion orientation analyzing algorithm based on Tibetan and Chinese mixed text. The experimental results showed that the method can obtain better performance in Tibetan and Chinese mixed Microblog orientation analysis.	No Mention

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[29]	An empirical study of sentiment analysis for Chinese documents	Four feature selection methods (MI, IG, CHI and DF) and five learning methods (centroid classifier, K-nearest neighbor, winnow classifier, Naïve Bayes and SVM) are investigated on a Chinese sentiment corpus with a size of 1021 documents. The experimental results indicate that IG performs the best for sentimental terms selection and SVM exhibits the best performance for sentiment classification. Furthermore, the authors found that sentiment classifiers are severely dependent on domains or topics.	No Mention
[30]	Adapting Information Bottleneck Method for Automatic Construction of Domain-oriented Sentiment Lexicon	The authors' theory verifies the convergence property of the proposed method. The empirical results also support the authors' theoretical analysis. In their experiment, it is shown that proposed method greatly outperforms the baseline methods in the task of building out-of-domain sentiment lexicon.	In this study, only the mutual information measure is employed to measure the three kinds of relationship. In order to show the robustness of the framework, the authors' future effort is to investigate how to integrate more measures into this framework.
[31]	Sentiment Classification for Consumer Word-of-Mouth in Chinese: Comparison between Supervised and Unsupervised Approaches	This study adopts three supervised learning approaches and a web-based semantic orientation approach, PMI-IR, to Chinese reviews. The results show that SVM outperforms naive bayes and N-gram model on various sizes of training examples, but does not obviously exceeds the semantic orientation approach when the number of training examples is smaller than 300.	No Mention
[32]	Modifying SO-PMI for Japanese Weblog Opinion Mining by Using a Balancing Factor and Detecting Neutral Expressions	After these modifications, the authors achieved a well-balanced result: both positive and negative Accuracy exceeded 70%. This shows that the authors' proposed approach not only adapted the SO-PMI for Japanese, but also modified it to analyze Japanese opinions more effectively.	In the future, the authors will evaluate different choices of words for the sets of positive and negative reference words. The authors also plan to appraise their proposal on other languages.
[33]	In this survey, the authors empirically evaluate the performance of different corpora in	Experiment results show that the Twitter data can achieve a much better performance than the Google, Web1T and Wikipedia based methods.	No Mention

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	sentiment similarity measurement, which is the fundamental task for word polarity classification.		
[34]	Adjective-Based Estimation of Short Sentence's Impression	The adjectives are ranked and top na adjectives are considered as an output of system. For example, the experiments were carried out and got fairly good results. With the input "it is snowy", the results are white (0.70), light (0.49), cold (0.43), solid (0.38), and scenic (0.37)	In the authors' future work, they will improve more in the tasks of keyword extraction and semantic similarity methods to make the proposed system working well with complex inputs.
[35]	JaYIIMard Index based Clustering Algorithm for Mining Online Review	In this work, the problem of predicting sales performance using sentiment information mined from reviews is studied and a novel JIBCA Algorithm is proposed and mathematically modeled. The outcome of this generates knowledge from mined data that can be useful for forecasting sales.	For future work, by using this framework, it can extend it to predicting sales performance in the other domains like customer electronics, mobile phones, computers based on the user reviews posted on the websites, etc.
[36]	Twitter sentiment classification for measuring public health concerns	Based on the number of tweets classified as Personal Negative, the authors compute a Measure of Concern (MOC) and a timeline of the MOC. We attempt to correlate peaks of the MOC timeline to the peaks of the News (Non-Personal) timeline. The authors' best Accuracy results are achieved using the two-step method with a Naïve Bayes classifier for the Epidemic domain (six datasets) and the Mental Health domain (three datasets).	No Mention
[37]	Ensemble of Classification algorithms for Subjectivity and Sentiment Analysis of Arabic Customers' Reviews	The experimental results show that the ensemble of the classifiers improves the classification effectiveness in terms of macro-F1 for both levels. The best results obtained from the subjectivity analysis and the sentiment classification in terms of macro-F1 are 97.13% and 90.95% respectively.	No Mention
[38]	Automatic Construction of Financial Semantic Orientation Lexicon from Large-Scale Chinese News	Semantic orientation lexicon of positive and negative words is indispensable for sentiment analysis. However, many lexicons are manually created by a small number of human subjects, which are susceptible to high cost and bias. In this survey, the authors propose a novel idea to construct a financial semantic orientation	No Mention

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	Corpus	lexicon from large-scale Chinese news corpus automatically ...	
[39]	Sentiment Classification in Under-Resourced Languages Using Graph-based Semi-supervised Learning Methods	In particular, the authors found that choosing initially labeled vertices in aYIIMordance with their degree and PageRank score can improve the performance. However, pruning unreliable edges will make things more difficult to predict. The authors believe that other people who are interested in this field can benefit from their empirical findings.	As future work, first, the authors will attempt to use a sophisticated approach to induce better sentiment features. The authors consider such elaborated features improve the classification performance, especially in the book domain. The authors also plan to exploit a much larger amount of unlabeled data to fully take advantage of SSL algorithms
[40]	A text-mining approach and combine it with semantic network analysis tools	In summary, the authors hope the text-mining and derived market-structure analysis presented in this paper provides a first step in exploring the extremely large, rich, and useful body of consumer data readily available on Web 2.0.	No Mention
[41]	Sentiment Classification in Resource-Scarce Languages by using Label Propagation	The authors compared our method with supervised learning and semi-supervised learning methods on real Chinese reviews classification in three domains. Experimental results demonstrated that label propagation showed a competitive performance against SVM or Transductive SVM with best hyper-parameter settings. Considering the difficulty of tuning hyper-parameters in a resourcescarce setting, the stable performance of parameter-free label propagation is promising.	The authors plan to further improve the performance of LP in sentiment classification, especially when the authors only have a small number of labeled seeds. The authors will exploit the idea of restricting the label propagating steps when the available labeled data is quite small.
[42]	A Vietnamese adjective emotion dictionary based on exploitation of Vietnamese language characteristics	The Vietnamese adjectives often bear emotion which values (or semantic scores) are not fixed and are changed when they appear in different contexts of these phrases. Therefore, if the Vietnamese adjectives bring sentiment and their semantic values (or their sentiment scores) are not changed in any context, then the results of the emotion classification are not high Accuracy. The authors propose many rules based on Vietnamese language characteristics to determine the emotional values of the Vietnamese adjective phrases bearing sentiment in specific contexts. The authors' Vietnamese sentiment adjective dictionary is widely used in applications and researches of the Vietnamese semantic classification.	not calculating all Vietnamese words completely; not identifying all Vietnamese adjective phrases fully, etc.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[43]	A Valences-Totaling Model for English Sentiment Classification	The authors present a full range of English sentences; thus, the emotion expressed in the English text is classified with more precision. The authors new model is not dependent on a special domain and training data set—it is a domain-independent classifier. The authors test our new model on the Internet data in English. The calculated valence (and polarity) of English semantic words in this model is based on many documents on millions of English Web sites and English social networks.	It has low Accuracy; it misses many sentiment-bearing English words; it misses many sentiment-bearing English phrases because sometimes the valence of a English phrase is not the total of the valences of the English words in this phrase; it misses many English sentences which are not processed fully; and it misses many English documents which are not processed fully.
[44]	Shifting Semantic Values of English Phrases for Classification	The results of the sentiment classification are not high Accuracy if the English phrases bring the emotions and their semantic values (or their sentiment scores) are not changed in any context. For those reasons, the authors propose many rules based on English language grammars to calculate the sentimental values of the English phrases bearing emotion in their specific contexts. The results of this work are widely used in applications and researches of the English semantic classification.	This survey is only applied to the English adverb phrases. The proposed model is needed to research more and more for the different types of the English words such as English noun, English adverbs, etc
[45]	A Valence-Totaling Model for Vietnamese Sentiment Classification	The authors have used the VTMfV to classify 30,000 Vietnamese documents which include the 15,000 positive Vietnamese documents and the 15,000 negative Vietnamese documents. The authors have achieved Accuracy in 63.9% of the authors' Vietnamese testing data set. VTMfV is not dependent on the special domain. VTMfV is also not dependent on the training data set and there is no training stage in this VTMfV. From the authors' results in this work, our VTMfV can be applied in the different fields of the Vietnamese natural language processing. In addition, the authors' TCMfV can be applied to many other languages such as Spanish, Korean, etc. It can also be applied to the big data set sentiment classification in Vietnamese and can classify millions of the Vietnamese documents	it has a low Accuracy.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[46]	Semantic Lexicons of English Nouns for Classification	The proposed rules based on English language grammars to calculate the sentimental values of the English phrases bearing emotion in their specific contexts. The results of the sentiment classification are not high Accuracy if the English phrases bring the emotions and their semantic values (or their sentiment scores) are not changed in any context. The valences of the English words (or the English phrases) are identified by using Tanimoto Coefficient (TC) through the Google search engine with AND operator and OR operator. The emotional values of the English noun phrases are based on the English grammars (English language characteristics)	This survey is only applied in the English noun phrases. The proposed model is needed to research more and more about the different types of the English words such as English English adverbs, etc.
Our work	-We use the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. The advantages and disadvantages of this survey are shown in the Conclusion section.		

Table 12: Comparisons of our model's results with the works related to the YULE-II coefficient (MC) in [58-63]

Studies	PMI	JM	YULE-II coefficient (MC)	Language	SD	DT	Sentiment Classification
[58]	Yes	Yes	Yes	English	NM	NM	No mention
[59]	No	No	Yes	NM	NM	NM	No mention
[60]	No	No	Yes	NM	NM	NM	No mention
[61]	No	No	Yes	NM	NM	NM	No mention
[62]	No	No	Yes	NM	NM	NM	No mention
[63]	No	No	Yes	NM	NM	NM	No mention
Our work	No	No	Yes	English Language	Yes	Yes	Yes

Table 12: Comparisons of our model's results with the works related to the YULE-II coefficient (MC) in [58-63]

Studies	PMI	JM	YULE-II coefficient (MC)	Language	SD	DT	Sentiment Classification
[58]	Yes	Yes	Yes	English	NM	NM	No mention
[59]	No	No	Yes	NM	NM	NM	No mention
[60]	No	No	Yes	NM	NM	NM	No mention
[61]	No	No	Yes	NM	NM	NM	No mention
[62]	No	No	Yes	NM	NM	NM	No mention

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[63]	No	No	Yes	NM	NM	NM	No mention
Our work	No	No	Yes	English Language	Yes	Yes	Yes

Table 13: Comparisons of our model's benefits and drawbacks with the studies related to the YULE-II coefficient (MC) in [58-63]

Surveys	Approach	Benefits	Drawbacks
[58]	A Survey of Binary Similarity and Distance Measures	Applying appropriate measures results in more accurate data analysis. Notwithstanding, few comprehensive surveys on binary measures have been conducted. Hence the authors collected 76 binary similarity and distance measures used over the last century and reveal their correlations through the hierarchical clustering technique	No mention
[59]	Yule and Kendall Coefficient	The Yule coefficient is used to measure the skewness of a frequency distribution. It takes into account the relative positions of the quartiles with respect to the median, and compares the spreading of the curve to the right and left of the median.	No mention
[60]	On Association Coefficients for 2×2 Tables and Properties That Do Not Depend on the Marginal Distributions	The authors study a family of coefficients that are linear transformations of the observed proportion of agreement given the marginal probabilities. This family includes the phi coefficient and Cohen's kappa. The main result is that the linear transformations that set the value under independence at zero and the maximum value at unity, transform all coefficients in this family into the same underlying coefficient. This coefficient happens to be Loevinger's H.	No mention
[61]	Statistical Inference for Generalized Yule Coefficients in 2 × 2 Contingency Tables	A confidence interval and sample size formula for a generalized Yule coefficient are proposed. The proposed confidence interval is shown to perform much better than the Wald intervals that are implemented in statistical packages.	No mention
[62]	This survey discusses common approaches to presenting the topic of skewness in the classroom, and explains why students need to know how to measure it	This study suggests reviving the Pearson 2 skewness statistic for the introductory statistics course because it compares the mean to the median in a precise way that students can understand. The research reiterates warnings about what any skewness statistic can actually tell the authors.	No mention
[63]	Data on the presence or absence of 25 fish species in a survey of 52 lakes from the watersheds of the Black and Hollow rivers of S-central Ontario were analyzed with 8 similarity coefficients	Measures of association and Ochiai's coefficient incorporate implicit centering transformations that reduce the size influence associated with the frequency of occurrence. Cluster analyses using co-occurrence coefficients are most susceptible to this size effect. The interpretations of many dendrograms fail to recognize size effects that arise from employing non-centered similarity coefficients. Arguments contrasting phenetic and phylogenetic methods may unknowingly debate the utility of centered versus non-centered coefficients, since the size effect undoubtedly contributes to the apparent strength of phylogenetic approaches	No mention
Our work	-We use the Self-Organizing Map algorithm, a testing data set and a training data set with the multi-dimensional vectors based on the sentiment lexicons of the bESD to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. The advantages and disadvantages of this survey are shown in the Conclusion section.		

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

APPENDIX OF CODES:

ALGORITHM 1: transferring one English document into one multi-dimensional vector in the sequential environment.

ALGORITHM 2: transferring all the documents of the testing data set into the multi-dimensional vectors in the sequential environment.

ALGORITHM 3: transferring all the positive documents of the training data set into all the multi-dimensional vectors, called the positive vector group of the training data set in the sequential system

ALGORITHM 4: transferring all the negative sentences of the training data set into all the one-dimensional vectors, called the negative vector group of the training data set in the sequential environment

ALGORITHM 5: creating one positive multi-dimensional central vector from the positive vector group of the training data set in the sequential system.

ALGORITHM 6: creating one negative multi-dimensional central vector from the negative vector group of the training data set in the sequential system.

ALGORITHM 7: setting the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the sequential system.

ALGORITHM 8:clustering all the documents of the testing data set into either the positive or the negative in the sequential system by using the SOM

ALGORITHM 9: implementing the Hadoop Map phase of transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system

ALGORITHM 10: implementing the Hadoop Reduce phase of transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system

ALGORITHM 11: implementing the Hadoop Map phase of transferring one document into the one-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system

ALGORITHM 12: implementing the Hadoop Reduce phase of transferring one document into the one-dimensional vectors of the document based on the sentiment lexicons of the bESD sentiment lexicons of the bESD in the parallel system.

ALGORITHM 13: performing the Hadoop Map phase of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

ALGORITHM 14: implementing the Hadoop Reduce phase of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

ALGORITHM 15: performing the Hadoop Map phase of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

ALGORITHM 16: implementing the Hadoop Reduce phase of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

ALGORITHM 17: implementing the Hadoop Map phase of creating one positive multi-dimensional central vector from the positive vector group of the training data set in the distributed system.

ALGORITHM 18: performing the Hadoop Reduce phase of creating one positive multi-dimensional central vector from the positive vector group of the training data set in the distributed system.

ALGORITHM 19: implementing the Hadoop Map phase of creating one negative multi-dimensional central vector from the negative vector group of the training data set in the distributed system.

ALGORITHM 20: performing the Hadoop Reduce phase of creating one negative multi-dimensional central vector from the negative vector group of the training data set in the distributed system.

ALGORITHM 21: performing the Hadoop Map phase of setting the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the distributed system.

ALGORITHM 22: implementing the Hadoop Reduce phase of setting the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the parallel system.

ALGORITHM 23: performing the Hadoop Map phase of using the Self-Organizing Map Algorithm (SOM) to cluster the documents into either the positive or the negative in the distributed system

ALGORITHM 24: implementing the Hadoop Reduce phase of using the Self-Organizing Map Algorithm (SOM) to cluster the documents into either the positive or the negative in the distributed system.

ALGORITHM 25: performing a basis English sentiment dictionary (bESD) in a sequential environment.

ALGORITHM 26: implementing the Hadoop Map phase of creating a basis English sentiment dictionary (bESD) in a distributed environment.

ALGORITHM 27: performing the Hadoop Reduce phase of creating a basis English sentiment dictionary (bESD) in a distributed environment.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ALGORITHM 1: transferring one English document into one multi-dimensional vector in the sequential environment.

Input: one English document
Output: the multi-dimensional vector
Begin
Step 1: Split the English document into many separate sentences based on "." Or "!" or "?";
Step 2: Set Multi-dimensionalVector := { } { } with n_max rows and m_max columns;
Step 3: Set i := 0;
Step 4: Each sentence in the sentences of this document, do repeat:
Step 5: Multi-dimensionalVector[i][] := {};
Step 6: Set j := 0;
Step 7: Split this sentence into the meaningful terms (meaningful words or meaningful phrases);
Step 8: Get the valence of this term based on the sentiment lexicons of the bESD;
Step 9: Add this term into Multi-dimensionalVector[i];
Step 10: Set j := j+ 1;
Step 11: End Repeat – End Step 4;
Step 12: While j is less than m_max, repeat:
Step 13: Add {0} into Multi-dimensionalVector[i];
Step 14: Set j := j+1;
Step 15: End Repeat – End Step 12;
Step 16: Set i := i+1;
Step 17: End Repeat – End Step 4;
Step 18: While i is less than n_max, repeat:
Step 19: Add the vector {0} into Multi-dimensionalVector;
Step 20: Set i := i+1;
Step 21: End Repeat – End Step 18;
Step 22: Return Multi-dimensionalVector;
End;

ALGORITHM 2: transferring all the documents of the testing data set into the multi-dimensional vectors in the sequential environment.

Input: the documents of the testing data set
Output: the multi-dimensional vectors of the testing data set
Begin
Step 1: Set TheMulti-dimensionalVectors := {}
Step 2: Each document in the documents of the testing data set, do repeat:
Step 3: OneMulti-dimensionalVector := the algorithm 4 to transfer one English document into one multi-dimensional vector in the sequential environment with the input is this document;
Step 4: Add OneMulti-dimensionalVector into TheMulti-dimensionalVectors;
Step 5: End Repeat- End Step 2;
Step 6: Return TheMulti-dimensionalVectors;
End;

ALGORITHM 3: transferring all the positive documents of the training data set into all the multi-dimensional vectors, called the positive vector group of the training data set in the sequential system

Input: all the positive documents of the training data set;
Output: the positive multi-dimensional vectors, called the positive vector group
Begin
Step 1: Set ThePositiveMulti-dimensionalVectors := null;
Step 2: Each document in the positive documents, repeat:
Step 3: Multi-dimensionalVector := the algorithm 4 to transfer one English document into one multi-dimensional vector in the sequential environment with the input is this document;
Step 4: Add Multi-dimensionalVector into ThePositiveMulti-dimensionalVectors;
Step 5: End Repeat – End Step 2;
Step 6: Return ThePositiveMulti-dimensionalVectors;
End;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ALGORITHM 4: transferring all the negative sentences of the training data set into all the one-dimensional vectors, called the negative vector group of the training data set in the sequential environment

Input: all the negative sentences of the training data set;
Output: the negative multi-dimensional vectors, called the negative vector group
Begin
Step 1: Set TheNegativeMulti-dimensionalVectors := null;
Step 2: Each document in the negative documents, repeat:
Step 3: Multi-dimensionalVector := the algorithm 4 to transfer one English document into one multi-dimensional vector in the sequential environment with the input is this document;
Step 4: Add Multi-dimensionalVector into TheNegativeMulti-dimensionalVectors;
Step 5: End Repeat – End Step 2;
Step 6: Return TheNegativeMulti-dimensionalVectors;
End;

ALGORITHM 5: creating one positive multi-dimensional central vector from the positive vector group of the training data set in the sequential system.

Input: ThePositiveOne-dimensionalVectors - the positive one-dimensional vectors, called the positive vector group
Output: one positive multi-dimensional central vector
Begin
Step 1: Set N := the number of the positive documents of the training data set;
Step 2: Set Multi-dimensionalCentralVector := null;
Step 3: For i := 0; i < n_max; i++, repeat : //rows
Step 4: Set One-dimensionalVector := null;
Step 5: Set Value := 0;
Setp 6: For j := 0; j < m_max; j++, repeat: //columns
Step 7: For k := 0; k < N; k++, repeat: //each Multi-dimensionalVector in the positive vector group
Step 8: Multi-dimensionalVector := ThePositiveOne-dimensionalVectors[k];
Step 9: Value := Value + Multi-dimensionalVector[i][j];
Step 10: End For – End Step 7;
Step 11: Add Value into One-dimensionalVector;
Step 12: End For – End Step 6;
Step 13: Add One-dimensionalVector into Multi-dimensionalCentralVector;
Step 14: End For – End Step 3;
Step 15: Return Multi-dimensionalCentralVector;
End;

ALGORITHM 6: creating one negative multi-dimensional central vector from the negative vector group of the training data set in the sequential system.

Input: TheNegativeOne-dimensionalVectors - the negative one-dimensional vectors, called the negative vector group
Output: one negative multi-dimensional central vector
Begin
Step 1: Set N := the number of the negative documents of the training data set;
Step 2: Set Multi-dimensionalCentralVector := null;
Step 3: For i := 0; i < n_max; i++, repeat : //rows
Step 4: Set One-dimensionalVector := null;
Step 5: Set Value := 0;
Setp 6: For j := 0; j < m_max; j++, repeat: //columns
Step 7: For k := 0; k < N; k++, repeat: //each Multi-dimensionalVector in the negative vector group
Step 8: Multi-dimensionalVector := TheNegativeOne-dimensionalVectors[k];
Step 9: Value := Value + Multi-dimensionalVector[i][j];
Step 10: End For – End Step 7;
Step 11: Add Value into One-dimensionalVector;
Step 12: End For – End Step 6;
Step 13: Add One-dimensionalVector into Multi-dimensionalCentralVector;
Step 14: End For – End Step 3;
Step 15: Return Multi-dimensionalCentralVector;
End;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ALGORITHM 7: setting the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the sequential system.

Input: one positive multi-dimensional central vector and one negative multi-dimensional central vector ; Matrix of the SOM

Output: the Matrix of the SOM;

Begin

Step 1: Set N := the n_max sentences of one document;
Step 2: Set PositiveOne-dimensionalCentralVector := null;
Step 3: Set NegativeOne-dimensionalCentralVector := null;
Step 4: For i := 0; i < m_max; i++, repeat://columns
Step 5: Set PositiveValue := 0;
Step 6: Set NegativeValue := 0;
Step 7: For j := 0; j < N; j++, repeat://rows
Step 8: PositiveValue := PositiveValue + the positive multi-dimensional central vector [j][i];
Step 9: NegativeValue := NegativeValue + the negative multi-dimensional central vector [j][i];
Step 10: End For – End Step 7;
Step 11: PositiveValue := (PositiveValue/N);
Step 12: NegativeValue := (NegativeValue/N);
Step 13: Add PositiveValue into PositiveOne-dimensionalCentralVector;
Step 14: Add NegativeValue into NegativeOne-dimensionalCentralVector;
Step 15: End For – End Step 4;
Step 16: Set the values of PositiveOne-dimensionalCentralVector for the first column of the Maxtrix of the SOM
Step 17: Set the values of NegativeOne-dimensionalCentralVector for the second column of the Maxtrix of the SOM
Step 18: Return The Matrix of the SOM

End;

ALGORITHM 8: clustering all the documents of the testing data set into either the positive or the negative in the sequential system by using the SOM

Input: the documents of the testing data set and the training data set

Output: positive, negative, neutral;

Begin

Step 0: the creating a basis English sentiment dictionary (bESD) in a sequential environment in (4.2.2);
Step 1: TheMulti-dimensionalVectors := the algorithm 5 to transfer all the documents of the testing data set into the multi-dimensional vectors in the sequential environment with the input is the documents of the testing data set;
Step 2: the algorithm 6 to transfer all the positive documents of the training data set into all the multi-dimensional vectors, called the positive vector group of the training data set in the sequential system
Step 3: the algorithm 7 to transfer all the negative documents of the training data set into all the multi-dimensional vectors, called the negative vector group of the training data set in the sequential environment.
Step 4: the algorithm 8 to create one positive multi-dimensional central vector from the positive vector group of the training data set in the sequential system
Step 5: the algorithm 9 to create one negative multi-dimensional central vector from the negative vector group of the training data set in the sequential system
Step 6: Set TheResultsOfTheSentimentClassification := {};
Step 7: Set Matrix := {}{} with its rows are the documents of the testing data set, the 2 columns
Step 8: Set i:= 0; and N := the documents of the testing data set;
Step 9: Each i in the 2 columns -1, do repeat:
Step 10: Set j := 0;
Step 11: the algorithm 10 to set the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the sequential system.
Step 12: Set Learning rate := 0.9;
Step 13: Set R := 0;
Step 14: While stopping condition false do step 15 to 21
Step 15: For each input vector x do step 16 to 18
Step 16: For ech j neuron, compute the Euclidean distance D(j)
Step 17: Find the index J such D(j) is a minimum
Step 18: For all neurons j within a specified neighbourhood of J and for all i: wji (new)= wij(old)+ learning rate * (xi - wij (old))

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 19: Update learning rate. It is a decreasing function of the number of epochs: learning rate (t+1) = [learning rate(t)]/2;
Step 20: Reduce radius of topolofical neighbourhood at specified times
Step 21: Test stop condition. Typically this is a small value of the learning rate with which the weight updates are insignificant.
Step 22: Set count_positive := 0 and count_negative := 0;
Step 23: Each j in the N -1, do repeat:
Step 24: If Matrix[j][0] is greater than Matrix[j][1] Then OneResult := positive;
Step 25: Else If Matrix[j][0] is less than Matrix[j][1] Then OneResult := negative;
Step 26: Else: OneResult := neutral;
Step 27: Add OneResult into TheResultsOfTheSentimentClassification;
Step 28: End Repeat – End Step 23;
Step 29: Return TheResultsOfTheSentimentClassification;
End;

ALGORITHM 9: implementing the Hadoop Map phase of the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system

Input: one document
Output: one one-dimensional vector
Begin
Step 1: Input this document into the Hadoop Map in the Cloudera system.
Step 2: Split this document into the sentences;
Step 3: Each sentence in the sentences, do repeat:
Step 4: One-dimensionalVector := null;
Step 5: Split this sentence into the meaningful terms;
Step 6: Each term in the meaningful terms, repeat:
Step 7: Get the valence of this term based on the sentiment lexicons of the bESD;
Step 8: Add this term into One-dimensionalVector;
Step 9 End Repeat – End Step 6;
Step 10: Return this One-dimensionalVector;
Step 11: The output of the Hadoop Map is this One-dimensionalVector;
End;

ALGORITHM 10: implementing the Hadoop Reduce phase of transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system

Input: One-dimensionalVector - one one-dimensional vector of the Hadoop Map (the input of the Hadoop Reduce is the output of the Hadoop Map)
Output: the multi-dimensional vector of the English document – Multi-dimensionalVector;
Begin
Step 1: Receive One-dimensionalVector;
Step 2: Add this One-dimensionalVector into One-dimensionalVector;
Step 3: Return Multi-dimensionalVector;
End;

ALGORITHM 11: implementing the Hadoop Map phase of transferring one document into the one-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system

Input: the documents of the testing data set
Output: one multi-dimensional vector (corresponding to one document)
Begin

Step 1: Input the documents of the testing data set into the Hadoop Map in the Cloudera system.
Step 2: Each document in the documents of the testing data set, do repeat:
Step 3: the multi-dimensional vector := the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system in Figure 12 with the input is this document;
Step 4: Return this multi-dimensional vector;
Step 5: The output of the Hadoop Map is this multi-dimensional vector;
End;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ALGORITHM 12: implementing the Hadoop Reduce phase of transferring one document into the one-dimensional vectors of the document based on the sentiment lexicons of the bESD sentiment lexicons of the bESD in the parallel system.

Input: one multi-dimensional vector of the Hadoop Map (the input of the Hadoop Reduce is the output of the Hadoop Map)

Output: the multi-dimensional vectors of the English documents of the testing data set

Begin

Step 1: Receive one multi-dimensional vector of the Hadoop Map

Step 2: Add this multi-dimensional vector into the multi-dimensional vectors of the testing data set;

Step 3: Return the multi-dimensional vectors of the testing data set;

End;

ALGORITHM 13: performing the Hadoop Map phase of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

Input: the positive documents of the training data set

Output: one multi-dimensional vector (corresponding to one document of the positive documents of the training data set)

Begin

Step 1: Input the positive documents into the Hadoop Map in the Cloudera system.

Step 2: Each document in the documents, do repeat:

Step 3: MultiDimentionalVector := the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system in Figure 12

Step 4: Return MultiDimentionalVector ;

End;

ALGORITHM 14: implementing the Hadoop Reduce phase of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

Input: one multi-dimensional vector (corresponding to one document of the positive documents of the training data set)

Output: the positive multi-dimensional vectors, called the positive vector group (corresponding to the positive documents of the training data set)

Begin

Step 1: Receive one multi-dimensional vector;

Step 2: Add this multi-dimensional vector into PositiveVectorGroup;

Step 3: Return PositiveVectorGroup - the positive multi-dimensional vectors, called the positive vector group (corresponding to the positive documents of the training data set);

End;

ALGORITHM 15: performing the Hadoop Map phase of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

Input: the negative documents of the training data set

Output: one multi-dimensional vector (corresponding to one document of the negative documents of the training data set)

Begin

Step 1: Input the negative documents into the Hadoop Map in the Cloudera system.

Step 2: Each document in the documents, do repeat:

Step 3: MultiDimentionalVector := the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system in Figure 12

Step 4: Return MultiDimentionalVector ;

End;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ALGORITHM 16: implementing the Hadoop Reduce phase of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

Input: one multi-dimensional vector (corresponding to one document of the negative documents of the training data set)

Output: the negative multi-dimensional vectors, called the negative vector group (corresponding to the negative documents of the training data set)

Begin

Step 1: Receive one multi-dimensional vector;
Step 2: Add this multi-dimensional vector into NegativeVectorGroup;
Step 3: Return NegativeVectorGroup - the negative multi-dimensional vectors, called the negative vector group (corresponding to the negative documents of the training data set);

End;

ALGORITHM 17: implementing the Hadoop Map phase of creating one positive multi-dimensional central vector from the positive vector group of the training data set in the distributed system.

Input: ThePositiveOne-dimensionalVectors - the positive one-dimensional vectors, called the positive vector group

Output: one positive multi-dimensional central vector

Begin

Step 1: Set N := the number of the positive documents of the training data set;
Step 2: Set Multi-dimensionalCentralVector := null;
Step 3: For i := 0; i < n_max; i++, repeat : //rows
Step 4: Set One-dimensionalVector := null;
Step 5: Set Value := 0;
Setp 6: For j := 0; j < m_max; j++, repeat: //columns
Step 7: For k := 0; k < N; k++, repeat: //each Multi-dimensionalVector in the positive vector group
Step 8: Multi-dimensionalVector := ThePositiveOne-dimensionalVectors[k];
Step 9: Value := Value + Multi-dimensionalVector[i][j];
Step 10: End For – End Step 7;
Step 11: Add Value into One-dimensionalVector;
Step 12: End For – End Step 6;
Step 13: Add One-dimensionalVector into Multi-dimensionalCentralVector;
Step 14: End For – End Step 3;
Step 15: Return Multi-dimensionalCentralVector;
Step 16: The output of the Hadoop Map is Multi-dimensionalCentralVector;

End;

ALGORITHM 18: performing the Hadoop Reduce phase of creating one positive multi-dimensional central vector from the positive vector group of the training data set in the distributed system.

Input: one positive multi-dimensional central vector – the output of the Hadoop Map

Output: one positive multi-dimensional central vector

Begin

Step 1: Receive one positive multi-dimensional central vector;
Step 2: Return one positive multi-dimensional central vector;

End;

ALGORITHM 19: implementing the Hadoop Map phase of creating one negative multi-dimensional central vector from the negative vector group of the training data set in the distributed system.

Input: TheNegativeOne-dimensionalVectors - the negative one-dimensional vectors, called the negative vector group

Output: one negative multi-dimensional central vector

Begin

Step 1: Set N := the number of the negative documents of the training data set;
Step 2: Set Multi-dimensionalCentralVector := null;
Step 3: For i := 0; i < n_max; i++, repeat : //rows
Step 4: Set One-dimensionalVector := null;
Step 5: Set Value := 0;
Setp 6: For j := 0; j < m_max; j++, repeat: //columns
Step 7: For k := 0; k < N; k++, repeat: //each Multi-dimensionalVector in the negative vector group

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 8: Multi-dimensionalVector := TheNegativeOne-dimensionalVectors[k];
Step 9: Value := Value + Multi-dimensionalVector[i][j];
Step 10: End For – End Step 7;
Step 11: Add Value into One-dimensionalVector;
Step 12: End For – End Step 6;
Step 13: Add One-dimensionalVector into Multi-dimensionalCentralVector;
Step 14: End For – End Step 3;
Step 15: Return Multi-dimensionalCentralVector;
Step 16: The output of the Hadoop Map is Multi-dimensionalCentralVector;
End;

ALGORITHM 20: performing the Hadoop Reduce phase of creating one negative multi-dimensional central vector from the negative vector group of the training data set in the distributed system.

Input: one negative multi-dimensional central vector – the output of the Hadoop Map
Output: one negative multi-dimensional central vector
Begin
Step 1: Receive one negative multi-dimensional central vector;
Step 2: Return one negative multi-dimensional central vector;
End;

ALGORITHM 21: performing the Hadoop Map phase of setting the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the distributed system.

Input: one positive multi-dimensional central vector and one negative multi-dimensional central vector ; Matrix of the SOM
Output: the Matrix of the SOM;
Begin
Step 0: Input one positive multi-dimensional central vector and one negative multi-dimensional central vector ; Matrix of the SOM into the Hadoop Map in the Cloudera system;
Step 1: Set N := the n_max sentences of one document;
Step 2: Set PositiveOne-dimensionalCentralVector := null;
Step 3: Set NegativeOne-dimensionalCentralVector := null;
Step 4: For i := 0; i < m_max; i++, repeat://columns
Step 5: Set PositiveValue := 0;
Step 6: Set NegativeValue := 0;
Step 7: For j := 0; j < N; j++, repeat://rows
Step 8: PositiveValue := PositiveValue + the positive multi-dimensional central vector [j][i];
Step 9: NegativeValue := NegativeValue + the negative multi-dimensional central vector [j][i];
Step 10: End For – End Step 7;
Step 11: PositiveValue := (PositiveValue/N);
Step 12: NegativeValue := (NegativeValue/N);
Step 13: Add PositiveValue into PositiveOne-dimensionalCentralVector;
Step 14: Add NegativeValue into NegativeOne-dimensionalCentralVector;
Step 15: End For – End Step 4;
Step 16: Set the values of PositiveOne-dimensionalCentralVector for the fist column of the Maxtrix of the SOM
Step 17: Set the values of NegativeOne-dimensionalCentralVector for the second column of the Maxtrix of the SOM
Step 18: Return the Maxtrix of the SOM;
End;

ALGORITHM 22: implementing the Hadoop Reduce phase of setting the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the parallel system.

Input: the Matrix of the SOM – the output of the Hadoop Map;
Output: the Matrix of the SOM;
Begin
Step 1:Receive the Matrix of the SOM;
Step 2: Return the Matrix of the SOM;
End;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ALGORITHM 23: performing the Hadoop Map phase of using the Self-Organizing Map Algorithm (SOM) to cluster the documents into either the positive or the negative in the distributed system

Input: the documents of the testing data set and the training data set
Output: positive, negative, neutral;
Begin
Step 0: the creating a basis English sentiment dictionary (bESD) in a distributed system in (4.2.3)
Step 1: TheMulti-dimensionalVectors := the transferring the documents of the testing data set into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system in Figure 13
Step 2: the transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system in Figure 14
Step 3: the transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative vector group of the training data set) based on the sentiment lexicons of the bESD] in the distributed system in Figure 15
Step 4: the creating one positive multi-dimensional central vector from the positive vector group of the training data set in the distributed system in Figure 16
Step 5: the creating one negative multi-dimensional central vector from the negative vector group of the training data set in the distributed system in Figure 17
Step 6: the setting the values of a center of one positive multi-dimensional central vector for the first column of Matrix of the SOM and the values of a center of one negative multi-dimensional central vector for the second column of Matrix of the SOM in the distributed system in Figure 18
Step 7: Input TheMulti-dimensionalVectors and the Matrix into the Hadoop Map in the Cloudera system;
Step 8: Set N := the documents of the testing data set;
Step 9: Set Learning rate := 0.9;
Step 10: Set R := 0;
Step 11: While stopping condition false do step 12 to 18
Step 12: For each input vector x do step 13 to 15
Step 13: For ech j neuron, compute the Euclidean distance D(j)
Step 14: Find the index J such D(j) is a minimum
Step 15: For all neurons j within a specified neighbourhood of J and for all i: wji (new)= wij(old)+ learning rate * (xi - wij (old))
Step 16: Update learning rate. It is a decreasing function of the number of epochs: learning rate (t+1) = [learning rate(t)]/2;
Step 17: Reduce radius of topolofical neighbourhood at specified times
Step 18: Test stop condition. Typically this is a small value of the learning rate with which the weight updates are insignificant.
Step 19: Set count_positive := 0 and count_negative := 0;
Step 20: Each j in the N -1, do repeat:
Step 21: If Matrix[j][0] is greater than Matrix[j][1] Then OneResult := positive;
Step 22: Else If Matrix[j][0] is less than Matrix[j][1] Then OneResult := negative;
Step 23: Else: OneResult := neutral;
Step 24: Return OneResult;
Step 25: The output of the Hadoop map is the OneResult;
End;

ALGORITHM 24: implementing the Hadoop Reduce phase of using the Self-Organizing Map Algorithm (SOM) to cluster the documents into either the positive or the negative in the distributed system.

Input: OneResult - the result of the sentiment classification of one document (the input of the Hadoop Reduce is the output of the Hadoop Map)
Output: the results of the sentiment classification of the documents of the testing data set;
Begin
Step 1: Receive OneResult - the result of the sentiment classification of one document
Step 2: Add this OneResult into the results of the sentiment classification of the documents of the testing data set;
Step 3: Return the results of the sentiment classification of the documents of the testing data set;
End;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ALGORITHM 25: performing a basis English sentiment dictionary (bESD) in a sequential environment.

Input: the 55,000 English terms; the Google search engine
Output: a basis English sentiment dictionary (bESD)
Begin
Step 1: Each term in the 55,000 terms, do repeat:
Step 2: By using eq. (8), eq. (9), and eq. (10) of the calculating a valence of one word (or one phrase) in English in the section (4.1.1), the sentiment score and the polarity of this term are identified. The valence and the polarity are calculated by using the YIIM through the Google search engine with AND operator and OR operator.
Step 3: Add this term into the basis English sentiment dictionary (bESD);
Step 4: End Repeat – End Step 1;
Step 5: Return bESD;
End;

ALGORITHM 26: implementing the Hadoop Map phase of creating a basis English sentiment dictionary (bESD) in a distributed environment.

Input: the 55,000 English terms; the Google search engine
Output: one term which the sentiment score and the polarity are identified.
Begin
Step 1: Each term in the 55,000 terms, do repeat:
Step 2: By using eq. (8), eq. (9), and eq. (10) of the calculating a valence of one word (or one phrase) in English in the section (4.1.1), the sentiment score and the polarity of this term are identified. The valence and the polarity are calculated by using the YIIM through the Google search engine with AND operator and OR operator.
Step 3: Return this term;
End;

ALGORITHM 27: performing thi Hadoop Reduce phase of creating a basis English sentiment dictionary (bESD) in a distributed environment.

Input: one term which the sentiment score and the polarity are identified – The output of the Hadoop Map phase.
Output: a basis English sentiment dictionary (bESD)
Begin
Step 1: Add this term into the basis English sentiment dictionary (bESD);
Step 2: Return bESD;
End;