Received: May 23, 2018

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Sentiment Classification in English Using a Self-Organizing Map Algorithm with Only the One-Dimensional Vectors and a Ruzicka Coefficient in a Parallel System

P. Vo Ngoc [1]* T. Vo Thi Ngoc [2]

[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
* Corresponding author's Email: vongocphu03hca@gmail.com

Abstract: Sentiment classification has been used in many different fields because it has many significant contributions in everyday life, such as in political activities, commodity production, and commercial activities. We have proposed a new model for big data sentiment classification by using a combination of an unsupervised learning algorithm of a machine learning with a Ruzicka Coefficient (RC) in this work. A Self-Organizing Map Algorithm (SOM) of the machine learning is used in clustering the documents of the testing data set (TES) comprising 7,500,000 documents, which are the 3,750,000 positive and the 3,750,000 negative in English, into either the positive group or the negative group of our training data set (TRA) which is 3,000,000 sentences including the 1,500,000 positive sentences and the 1,500,000 negative sentences in English. In this study, we do not use a vector space modeling (VSM). We do not use any multi-dimensional vectors according to both the VSM and many sentiment lexicons. We use many sentiment lexicons of our basis English sentiment dictionary (bESD). We use many one-dimensional vectors based on the sentiment lexicons. We use a similarity coefficient in this study. We do not use any one-dimensional vectors based on the VSM. We have achieved 88.64% accuracy of the TES. The execution time of the proposed model in a distributed network environment - DNE is less than that in a sequential system - SS. Many commercial applications and surveys of the sentiment classification can widely use the results of the proposed model.

Keywords: English sentiment classification, Distributed system, Cloudera, Hadoop map and Hadoop reduce, Ruzicka coefficient, Self-organizing map.

1. Introduction

A Self-Organizing Map Algorithm (SOM) has been displayed in more details in the researches [1 - 3].

The basic principles of the proposed model are as follows: (1) Assuming that each English sentence has m English words (or English phrases). (2) Assuming 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) Assuming 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 a Ruzicka Coefficient (RC) 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 and originality of the proposed approach are presented: (1) The RC was applied to sentiment analysis. (2) This algorithm can also be applied to identify the emotions of millions of documents. The SOM was applied to the sentiment classification. (3) This survey can be applied to other parallel network systems. (4) The Cloudera distributed network system (CDNS), Hadoop Map (M) and Hadoop Reduce (R) were used in the proposed model. (5) We did not use the VSM [4-6]. (6) We used many sentiment lexicons. (7) We did not use any multi-dimensional vectors according to both the VSM and the sentiment lexicons. (8) We used many one-dimensional vectors based on the sentiment lexicons. (9) We did not use any one-dimensional vectors according to the VSM. (10) We used a RC through a Google search engine with AND operator and OR operator to identify many sentiment values and polarities of the sentiment lexicons in English. (11) We used the SOM to classify one document of the testing data set - TES into either the positive polarity or the negative polarity. (12) The input of this survey is the documents of the TES and the sentences of the training data set - TRA in English. We studied to transfer the documents and the sentences into the formats for the SOM which can process them. (13) We tested the proposed model in both a sequential environment - SE and a distributed network system - DNE. (14) We built the RC – related equations and the SOM – related algorithms in this survey. (15) To identify the sentiment classification of all the documents of the TES, the SOM was implemented many times. (16) We surveyed to calculate one positive one-dimensional central vector of all the positive sentences and one negative one-dimensional central vector of all the negative sentences of the TRA. Then, we studied to set the value of a matrix (a Map) of the SOM successfully. (17) We built the algorithms related to the SOM and the RC in the PNE. Therefore, we have studied this model in more details.

We perform the proposed model as follows:

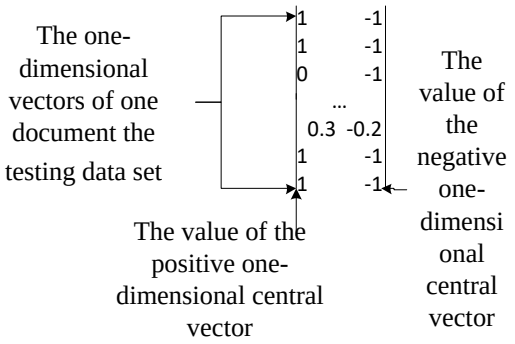

Figure. 1 An initialization of the SOM – the Map

We transfer all the sentences of one document of the TES into the all the one-dimensional vectors of the document based on the sentiment lexicons of the bESD. We transfer all the positive sentences of the TRA into the one-dimensional vectors based on the sentiment lexicons, called the positive vector group of the TRA. One positive one-dimensional central vector is identified from the positive vector group. We transfer all the negative sentences of the TRA into the one-dimensional vectors based on the sentiment lexicons, called the negative vector group of the TRA. One negative one-dimensional central vector is identified from the negative vector group. One document of the TES is transferred into the one-dimensional vectors of the document based on the sentiment lexicons of the bESD. The one-dimensional vectors of one document are clustered into either the positive group or the negative group by using the SOM with the input is all the one-dimensional vectors of this document. We set an initialization of the SOM with its map in Fig. 1.

The Map of the SOM is a matrix which has n_max rows and 2 columns. The n_max is the number of sentences of this document of the TES. The positive one-dimensional central vector is used in setting the values for the values of the first column of the Map and the negative one-dimensional central vector is used in setting the values for the values of the second column of the Map. Then, after the SOM is implemented completely, we have the Map in Fig. 2. In Fig. 2, we have the one-dimensional vector 1 (0.1, 0), the one-dimensional vector 2 (0, 0.2), the one-dimensional vector 3 (0.3, 0.3), the one-dimensional vector 4 (0.4, 0.1), and the one-dimensional vector 5 (0.2, 0.5) as follows: (1) With the one-dimensional vector 1 (corresponding to the sentence 1 of the document the TES), 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

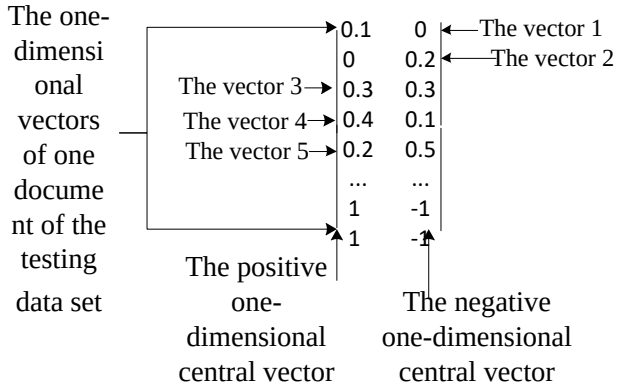

Figure. 2 The final Map – the result of clustering by using the SOM

greater than the value of the column of the negative polarity, therefore this one-dimensional vector is clustered into the positive. (2) With the one-dimensional vector 2 (corresponding to the sentence 2 of the document of the TES), 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 one-dimensional vector is clustered into the negative. (3) With the one-dimensional vector 3 (corresponding to the sentence 3 of the document of the TES), 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 one-dimensional vector is clustered into the neutral polarity. (4) With the one-dimensional vector 4 (0.4, 0.1) (corresponding to the sentence 4 of the document the TES), 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 one-dimensional vector is clustered into the positive. (5) With the one-dimensional vector 5 (corresponding to the sentence 5 of the document of the TES), 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 one-dimensional vector is clustered into the negative.

Finally, the sentiment classification of all the sentences of the document of the TES is identified completely. The sentiment classification of this document is identified according to the results of the sentiment classification of its sentences. This document is clustered into the positive if the number of the sentences clustered into the positive group is greater than the sentences clustered into the negative group in the document. This document is clustered into the negative if the number of the sentences clustered into the positive group is less than the sentences clustered into the negative group in the document. This document is clustered into the neutral if the number of the sentences clustered into the positive group is as equal as the sentences clustered into the negative group in the document. Finally, the sentiment classification of all the documents of the TES is identified completely.

We firstly perform all the above things in the sequential system - SS 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. We secondly perform all the above things in the parallel network environment – PNE secondly 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 strong points of this proposed method are displayed: using the SOM, the RC, the sentiment lexicons, the one-dimensional vectors, the sentences of the TRA, not using the VSM, not using the multi-dimensional vectors, and using the one-dimensional vectors of the sentiment lexicons, etc.

The new features of the proposed model and the main advantages of the results have been better than the researches related to the same topic of this novel model. The results of this model have outperformed a lot.

Our proposed model has the significant contributions which 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. (2) The algorithms are built in the proposed model. (3) This survey can certainly be applied to other languages easily. (4) The results of this study can significantly be applied to the types of other words in English. (5) The algorithm of data mining is applicable to semantic analysis of natural language processing. (6) This study also proves that different fields of scientific research can be related in many ways. (7) Millions of English documents are successfully processed for emotional analysis. (8) The semantic classification is implemented in the parallel network environment. (9) The principles are proposed in the research. (10) The Cloudera distributed environment is used in this study. (11) The proposed work can be applied to other distributed systems. (12) This survey uses M and R. (13) Our proposed model can be applied to many different parallel network environments such as a Cloudera system. (14) This study can be applied to many different distributed functions such as M and R. (15) The SOM – related algorithms are proposed in this survey. (16) The RC – 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), Ruzicka Coefficient (RC), 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 Reference section comprises all the reference documents.

2. Related work

We summarize many researches which are related to our research. There are the researches related to the Self-Organizing Map Algorithm (SOM) in [1-3]. In [1], the self-organized map, one architecture suggested for artificial neural networks, was explained by presenting simulation experiments and practical applications. The self-organizing map had the property of effectively creating spatially organized internal representations of various features of input signals and their abstractions, etc.

There are the works related to vector space modeling (VSM) in [4-6]. In this study [4], the authors examined the VSM, an Information Retrieval technique and its variation, etc.

The latest researches of the sentiment classification are [7-13]. In the research [7], the authors presented 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, 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 Ruzicka coefficient (RC) 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 [14-16]. In the research [14], the authors generated several Norwegian sentiment lexicons by extracting sentiment information from two different types of Norwegian text corpus, namely, news corpus and discussion forums, etc.

Two studies related to the PMI measure and Jaccard measure are in [17, 18]. In the survey [17], the authors empirically evaluated the performance of different corpora in sentiment similarity measurement, which was the fundamental task for word polarity classification, etc.

The works related to the Jaccard measure are in [19-21]. The survey in [19] investigated the problem of sentiment analysis of the online review, etc.

The surveys related to the similarity coefficients to calculate the valences of words are in [22-26]. It has taken lots of cost and time to do this work [22].

The English dictionaries are [27-29] 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 Ruzicka coefficient (RC) in [30-32]. The authors in [30] collected 76 binary similarity and distance measures used over the last century and reveal their correlations through the hierarchical clustering technique, etc.

This research [30] has had many cost and time to be implemented well, etc.

3. Data set

We built our TES including the 7,500,000 documents in the movie field, which contains the 3,750,000 positive and 3,750,000 negative in English. We also built our TRA including the 3,000,000 sentences in the movie field, which contains the 1,500,000 positive and 1,500,000 negative in English. All the documents of our TES and all the sentences of our TRA are automatically extracted from English Facebook, English websites and social networks; then we labeled positive and negative for them.

4. Methodology

There are two sub-sections in this section: 4.1 and 4.2.

4.1 Using the SOM to cluster the documents of the TES into either the positive or the negative in both a SE and a DNE

In Fig. 3, an overview of the proposed model is presented. There are two sub-sections of this section: 4.1.1 and 4.1.2.

4.1.1. Using the SOM to cluster the documents of the TES into either the positive or the negative in a SE

The SOM is used in clustering the documents of the TES into either the positive or the negative in a SE in this section.

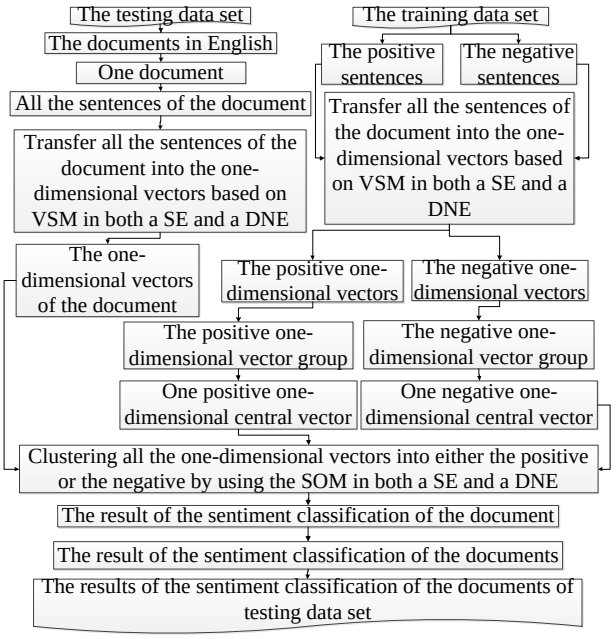

Figure. 3 Overview of our new model

We proposed the algorithm 1 to transfer one English sentence into the one-dimensional vector based on the sentiment lexicons of the bESD in the SE
Input: one English sentence
Output: one one-dimensional vector
Step 1: Split this sentence into the meaningful terms
Step 2: One-dimensionalVector := null;
Step 3: Each term in the terms, repeat:
Step 4: Get the sentiment value of this term based on the sentiment lexicons of the bESD;
Step 5: Add this term into One-dimensionalVector;
Step 6: End Repeat-End Step 3;
Step 2: Return One-dimensionalVector;

We built the algorithm 2 to transfer one English document into the one-dimensional vectors of the document in the SE
Input: one English document
Output: the one-dimensional vectors of this document
Step 1: Split the English document into many separate sentences based on "." Or "!" or "?";
Step 2: Set TheOne-dimensionalVectors := null;
Step 3: Each sentence in the sentences of this document, do repeat:
Step 4: One-dimensionalVector := the algorithm 4 to transfer one English sentence into the one-dimensional vector based on the sentiment lexicons of the bESD in the SE with the input is this sentence;
Step 5: Add One-dimensionalVector into TheOne-dimensionalVectors;
Step 6: End Repeat – End Step 2
Step 7: Return TheOne-dimensionalVectors;
with:
TheOne-dimensionalVectors is an array of one-dimensional vectors of the document (one document)
One-dimensionalVector is one one-dimensional vector of a sentence.

We developed the algorithm 3 to transfer all the positive sentences of the TRA into all the one-dimensional vectors, called the positive vector group of the TRA in the SS
Input: all the positive sentences of the TRA;
Output the positive one-dimensional vectors, called the positive vector group
Step 1: Set ThePositiveOne-dimensionalVectors := null;
Step 2: Each sentence in the positive sentences, repeat:
Step 3: One-dimensionalVector := the algorithm 4 to transfer one English sentence into the one-dimensional vector based on the sentiment lexicons of the bESD in the SE with the input is this sentence;
Step 4: Add One-dimensionalVector into ThePositiveOne-dimensionalVectors;
Step 5: End Repeat – End Step 2;
Step 6: Return ThePositiveOne-dimensionalVectors;
with:
ThePositiveOne-dimensionalVectors is a positive one-dimensional vector group (positive one-dimensional vectors) of the positive sentences of TRA

We implemented the algorithm 4 to transfer all the negative sentences of the TRA into all the one-dimensional vectors, called the negative vector group of the TRA in the SE
Input: all the negative sentences of the TRA;
Output the negative one-dimensional vectors, called the negative vector group
Step 1: Set TheNegativeOne-dimensionalVectors := null;
Step 2: Each sentence in the negative sentences, repeat:
Step 3: One-dimensionalVector := the algorithm 1 to transfer one English sentence into the one-dimensional vector based on the sentiment lexicons of the bESD in the SE with the input is this sentence;
Step 4: Add One-dimensionalVector into TheNegativeOne-dimensionalVectors;
Step 5: End Repeat – End Step 2;
Step 6: Return TheNegativeOne-dimensionalVectors;

with:
TheNegativeOne-dimensionalVectors is a negative one-dimensional vector group (negative one-dimensional vectors) of the negative sentences of TRA

We performed the algorithm 5 to create one positive one-dimensional central vector from the positive vector group of the TRA in the SE
Input: ThePositiveOne-dimensionalVectors - the positive one-dimensional vectors, called the positive vector group
Output: one positive one-dimensional central vector
Step 1: Set N := the number of the positive sentences of the TRA;
Step 2: Set One-dimensionalVector := null;
Step 3: For i := 0; i < m_max, repeat:
Step 4: Set Value := 0;
Step 5: Each vector in the positive vector group, repeat:
Step 6: Value := Value + vector [i];
Step 7: End Repeat – End Step 5;
Step 8: Value := (Value /N);
Step 9: Add Value into One-dimensionalVector;
Step 10: End For – End Step 3;
Step 11: Return One-dimensionalVector;
with:
One-dimensionalVector is one positive one-dimensional central vector

We developed the algorithm 6 to create one negative one-dimensional central vector from the negative vector group of the TRA in the SE
Input: TheNegativeOne-dimensionalVectors - the negative one-dimensional vectors, called the negative vector group
Output: one negative one-dimensional central vector
Step 1: Set N := the number of the negative sentences of the TRA;
Step 2: Set One-dimensionalVector := null;
Step 3: For i := 0; i < m_max, repeat:
Step 4: Set Value := 0;
Step 5: Each vector in the negative vector group, repeat:
Step 6: Value := Value + vector [i];
Step 7: End Repeat – End Step 5;
Step 8: Value := (Value /N);
Step 9: Add Value into One-dimensionalVector;
Step 10: End For – End Step 3;
Step 11: Return One-dimensionalVector;
with:
One-dimensionalVector is one negative one-dimensional central vector

We proposed the algorithm 7 to cluster one document of the TES into either the positive or the negative by using the SOM in the SE
Input: one document; one positive one-dimensional vector and one negative one-dimensional vector;
Output: positive, negative, neutral;
Step 1: Set Matrix := {}{} with the n_max rows, the 2 columns
Step 2: Set i:= 0;
Step 3: Set the values of the positive one-dimensional vector for the values of the column 0 of Matrix
Step 4: Set the values of the negative one-dimensional vector for the values of the column 1 of Matrix
Step 5: Set Learning rate := 0.9;
Step 6: Set R := 0;
Step 7: While stopping condition false do step 8 to 14
Step 8: For each input vector x do step 9 to 11
Step 9: For ech j neuron, compute the Euclidean distance D(j)
Step 10: Find the index J such D(j) is a minimum
Step 11: For all neurons j within a specified neighbourhood of J and for all i: wji (new)= wij(old)+ learning rate * (xi - wij (old))
Step 12: Update learning rate. It is a decreasing function of the number of epochs: learning rate (t+1) = [learning rate(t)]/2;
Step 13: Reduce radius of topolofical neighbourhood at specified times
Step 14: Test stop condition. Typically this is a small value of the learning rate with which the weight updates are insignificant.
Step 15: Set count_positive := 0 and count_negative := 0;
Step 16: Ech j in the n_max rows -1, do repeat:
Step 17: If Matrix[j][0] is greater than Matrix[j][1] Then count_positive := count_positive +1;
Step 18: If Matrix[j][0] is less than Matrix[j][1] Then count_negative := count_negative +1;
Step 19: End Repeat – End Step 16;
Step 20: If count_positive is greater than count_negative Then Return positive;
Step 21: Else If count_positive is less than count_negative Then Return negative;
Step 22: Return neutral;

We built the algorithm 8 to cluster the documents of the TES into either positive or the negative by using the SOM in the SS
Input: the TES and TRA;
Output: the results of the sentiment classification of the TES

Step 1: Set TheResults := null; and The creating a bESD in a sequential environment (4.2.2)
Step 2: The algorithm 3 to transfer all the positive sentences of the TRA into all the one-dimensional vectors, called the positive vector group of the TRA in the SS with the input is the positive sentences of the TRA;
Step 3: The algorithm 4 to transfer all the negative sentences of the TRA into all the one-dimensional vectors, called the negative vector group of the TRA in the SS with the input is the negative sentences of the TRA;
Step 4: The algorithm 5 to create one positive one-dimensional central vector from the positive vector group of the TRA in the SS with the input is the positive vector group;
Step 5: The algorithm 6 to create one negative one-dimensional central vector from the negative vector group of the TRA in the SE with the input is the negative vector group;
Step 6: Each document in the documents of the TES, do repeat:
Step 7: OneResult := the algorithm 7 to cluster one document of the TES into either the positive or the negative by using the SOM in the SE with the input is this document;
Step 8: Add OneResult into TheResults;
Step 9: End Repeat – End Step 6;
Step 10: Return TheResults;

4.1.2. Using the SOM to cluster the documents of the TES into either the positive or the negative in a DNE

The SOM is used in clustering the documents of the TES into either the positive or the negative in a DNE in this part.

We transformed one English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in CDNS in the algorithm 9 and the algorithm 10. This stage includes two phases: the M phase and the R phase. The input of the M is one sentence and the bESD. The output of the M is one term (one meaningful word/or one meaningful phrase) which the valence is identified. The input of the R is the output of the M, thus, the input of the R is one term (one meaningful word/or one meaningful phrase) which the valence is identified. The output of the R is one one-dimensional vector of this sentence.

We developed the algorithm 9 to perform the M:
Input: one sentence and the bESD;
Output: one term (one meaningful word/or one meaningful phrase) which the valence is identified
Step 1: Input this sentence and the bESD into the M in the CDNS;
Step 2: Split this sentence into the many meaningful terms (meaningful words/or meaningful phrases) based on the bESD;
Step 3: Each term in the terms, do repeat:
Step 4: Identify the valence of this term based on the bESD;
Step 5: Return this term; //the output of the M.

We proposed the algorithm 10 to perform the R:
Input: one term (one meaningful word/one meaningful phrase) which the valence is identified – the output of the M
Output: one one-dimensional vector based on the sentiment lexicons of the bESD
Step 1: Receive one term;
Step 2: Add this term into the one-dimentional vector;
Step 3: Return the one-dimentional vector;

We transferred one document into the one-dimensional vectors of the document based on the sentiment lexicons of the bESD in the PNE in the algorithm 11 and the algorithm 12. This stage includes two phases: the M phase and the R phase. The input of the M is one document. The output of the M is one one-dimensional vector. The input of the R is the M, thus, the input of the R is one one-dimensional vector. The output of the R is the one-dimensional vectors of this document.

We proposed the algorithm 11 to implement the M:
Input: one document;
Output: one one-dimensional vector (corresponding to one sentence)
Step 1: Input this document into the M in the CDNS
Step 2: Split this document into the n sentences;
Step 3: Each sentence in the n sentences, do repeat:
Step 4: the one-dimensional vector := transferring one sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in the CDNS in the algorithm 9 and the algorithm 10 with the input is this sentence;
Step 5: Return this one-dimensional vector;
Step 6: The output of the M is this one-dimensional vector;

We developed the algorithm 12 to implement the R:
Input: one one-dimensional vector of the M (the input of the R is the output of the M)
Output: the one-dimensional vectors of the English document
Step 1: Receive one one-dimensional vector of the M

Step 2: Add this one-dimensional vector into the one-dimensional vectors of the English document
Step 3: Return the one-dimensional vectors of the English document;

We transformed the positive sentences of the TRA into the positive one-dimensional vectors (called the positive vector group of the TRA) in the DNE in the algorithm 13 and the algorithm 14. The stage includes two phases: the M phase and the R phase. The input of the M is the positive sentences of the TRA. The output of the M is one one-dimensional vector of the positive sentences of the TRA. The input of the R is the output of the M, thus, the input of the R is one one-dimensional vector of one sentence of the positive sentences of the TRA). The output of the R is the positive one-dimensional vectors, called the positive vector group (corresponding to the positive sentences of the TRA)

We built the algorithm 13 to perform the M:
Input: the positive sentences of the TRA
Output: one one-dimensional vector of the positive sentences of the TRA
Step 1: Input the positive sentences into the M in the Cloudera system.
Step 2: Each sentences in the positive sentences, do repeat:
Step 3: OneOne-DimentionalVector := transferring one sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in the CDNS in the algorithm 9 and the algorithm 10.
Step 4: Return OneOne-DimentionalVector ;

We proposed the algorithm 14 to implement the R:
Input: one one-dimensional vector of the positive sentences of the TRA
Output: the positive one-dimensional vectors, called the positive vector group (corresponding to the positive sentences of the TRA)
Step 1: Receive one one-dimensional vector;
Step 2: Add this one-dimensional vector into PositiveVectorGroup;
Step 3: Return PositiveVectorGroup - the positive one-dimensional vectors, called the positive vector group (corresponding to the positive sentences of the TRA);

We transferred the negative sentences of the TRA into the negative one-dimensional vectors (called the negative vector group of the TRA) in the PNE in the algorithm 15 and the algorithm 16. The stage includes two phases: the M phase and the R phase. The input of the M is the negative sentences of the TRA. The output of the M is one one-dimensional vector of the negative sentences of the TRA. The input of the R is the output of the M, thus, the input of the R is one one-dimensional vector of one sentence of the negative sentences of the TRA). The output of the R is the positive one-dimensional vectors, called the negative vector group (corresponding to the negative sentences of the TRA)

We built the algorithm 15 to perform the M:
Input: the negative sentences of the TRA
Output: one one-dimensional vector of the negative sentences of the TRA
Step 1: Input the negative sentences into the M in the Cloudera system.
Step 2: Each sentences in the negative sentences, do repeat:
Step 3: OneOne-DimentionalVector := transferring one sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in the DNE in the algorithm 9 and the algorithm 10.
Step 4: Return OneOne-DimentionalVector ;

We proposed the algorithm 16 to implement the R:
Input: one one-dimensional vector of the negative sentences of the TRA
Output: the negative one-dimensional vectors, called the negative vector group (corresponding to the negative sentences of the TRA)
Step 1: Receive one one-dimensional vector;
Step 2: Add this one-dimensional vector into NegativeVectorGroup;
Step 3: Return NegativeVectorGroup - the negative one-dimensional vectors, called the negative vector group (corresponding to the negative sentences of the TRA);

We created one positive one-dimensional central vector from the positive vector group of the TRA in the DNE in the algorithm 17 and the algorithm 18. The stage includes two phases: the M phase and the R phase. The input of the M is the positive one-dimensional vectors, called the positive vector group. The output of the M is one positive one-dimensional central vector. The input of the R is the output of the M, thus, the input of the R is one positive one-dimensional central vector. The output of the R is one positive one-dimensional central vector

We performed the algorithm 17 to implement the M:

Input: ThePositiveOne-dimensionalVectors - the positive one-dimensional vectors, called the positive vector group
Output: one positive one-dimensional central vector
Step 1: Set N := the number of the positive sentences of the TRA;
Step 2: Set One-dimensionalVector := null;
Step 3: For i := 0; i < m_max, repeat:
Step 4: Set Value := 0;
Step 5: Each vector in the positive vector group, repeat:
Step 6: Value := Value + vector [i];
Step 7: End Repeat – End Step 5;
Step 8: Value := (Value /N);
Step 9: Add Value into One-dimensionalVector;
Step 10: End For – End Step 3;
Step 11: Return One-dimensionalVector;
Step 12: The output of the Hadoop Map is One-dimensionalVector

We proposed the algorithm 18 to perform the R:
Input: one positive one-dimensional central vector
Output: one positive one-dimensional central vector
Step 1: Receive one positive one-dimensional central vector;
Step 2: Return one positive one-dimensional central vector;

We created one negative one-dimensional central vector from the negative vector group of the TRA in the DNE in the algorithm 19 and the algorithm 20. The stage includes two phases: the M phase and the R phase. The input of the M is the negative one-dimensional vectors, called the negative vector group. The output of the M is one negative one-dimensional central vector. The input of the R is the output of the M, thus, the input of the R is one negative one-dimensional central vector. The output of the Hadoop Reduce phase is one negative one-dimensional central vector

We developed the algorithm 19 to implement the M:
Input: TheNegativeOne-dimensionalVectors - the negative one-dimensional vectors, called the negative vector group
Output: one negative one-dimensional central vector
Step 1: Set N := the number of the negative sentences of the TRA;
Step 2: Set One-dimensionalVector := null;
Step 3: For i := 0; i < m_max, repeat:
Step 4: Set Value := 0;
Step 5: Each vector in the negative vector group, repeat:
Step 6: Value := Value + vector [i];
Step 7: End Repeat – End Step 5;
Step 8: Value := (Value /N);
Step 9: Add Value into One-dimensionalVector;
Step 10: End For – End Step 3;
Step 11: Return One-dimensionalVector;
Step 12: The output of the Hadoop Map is One-dimensionalVector

We proposed the algorithm 20 to perform the R:
Input: one negative one-dimensional central vector
Output: one negative one-dimensional central vector
Step 1: Receive one negative one-dimensional central vector;
Step 2: Return one negative one-dimensional central vector;

We used the SOM to cluster one document into either the positive or the negative in the DNE in the algorithm 21 and the algorithm 22. This stage comprises two phases: the M phase and the R phase. The input of the M is one document of the TES, one positive one-dimensional central vector and one negative one-dimensional central vector. The output of the M is the result of the sentiment classification of this document. The input of the R is the output of the M, thus, the input of the Hadoop Reduce is the result of the sentiment classification of this document. The output of the R is the result of the sentiment classification of this document

We performed the algorithm 21 to perform the M:
Input: one document; one positive one-dimensional central vector and one negative one-dimensional central vector
Output: the result of the sentiment classification of this document
Step 1: Set Matrix := {}{} with the n_max rows, the 2 columns
Step 2: Set the values of the positive one-dimensional central vector for the values of the column 0 of Matrix;
Step 3: Set the values of the negative one-dimensional central vector for the values of the column 1 of Matrix;
Step 4: Set Learning rate := 0.9;
Step 5: Set R := 0;
Step 6: While stopping condition false do step 7 to 13
Step 7: For each input vector x do step 8 to 10
Step 8: For ech j neuron, compute the Euclidean distance D(j)
Step 9: Find the index J such D(j) is a minimum

Step 10: For all neurons j within a specified neighbourhood of J and for all i: wji (new)= wij(old)+ learning rate * (xi - wij (old))
Step 11: Update learning rate. It is a decreasing function of the number of epochs: learning rate (t+1) = [learning rate(t)]/2;
Step 12: Reduce radius of topolofical neighbourhood at specified times
Step 13: Test stop condition. Typically this is a small value of the learning rate with which the weight updates are insignificant.
Step 14: Set count_positive := 0 and count_negative := 0;
Step 15: Ech j in the n_max rows -1, do repeat:
Step 17: If Matrix[j][0] is greater than Matrix[j][1] Then count_positive := count_positive +1;
Step 18: If Matrix[j][0] is less than Matrix[j][1] Then count_negative := count_negative +1;
Step 19: End Repeat – End Step 15;
Step 20: If count_positive is greater than count_negative Then Return positive;
Step 21: Else If count_positive is less than count_negative Then Return negative;
Step 22: Return neutral;

We proposed the algorithm 22 to implement the R:
Output: the result of the sentiment classification of this document
Step 1: Receive the result of the sentiment classification of this document
Step 2: Return the result of the sentiment classification of this document;

We clustered the documents of the TES into either positive or the negative by using the SOM in the DNE in the algorithm 23 and the algorithm 24. This stage includes two phases: the M phase and the R phase. The input of the M is the documents of the TES and the TRA. The output of the M is one result of the sentiment classification of one document of the TES. The input of the R is the M, thus, the input of the R is one result of the sentiment classification of one document of the TES. The output of the R is the results of the sentiment classification of the TES

We built the algorithm 23 to implement the M:
Input: the documents of the TES and the TRA;
Output: one result of the sentiment classification of one document of the TES;
Step 0: the creating a bESD in a DNE (4.2.3)
Step 1: transferring the positive sentences of the TRA into the positive one-dimensional vectors (called the positive vector group of the TRA) in the DNE in the algorithm 13 and the algorithm 14.
Step 2: transferring the negative sentences of TRA into the negative one-dimensional vectors (called the negative vector group of the TRA) in the DNE in in the algorithm 15 and the algorithm 16.
Step 3: creating one positive one-dimensional central vector from the positive vector group of the TRA in the DNE in the algorithm 17 and the algorithm 18.
Step 4: creating one negative one-dimensional central vector from the negative vector group of the TRA in the DNE in in the algorithm 19 and the algorithm 20
Step 5: Each document of the documents of the TES, do repeat:
Step 6: OneResult := using the SOM to cluster one document into either the positive or the negative in the DNE in the algorithm 21 and the algorithm 22.
Step 7: Return this OneResult;
Step 8: The output of the Hadoop Map is this OneResult;

We proposed the algorithm 24 to implement the R:
Input: one result of the sentiment classification of one document of the TES;
Output: the results of the sentiment classification of the TES;
Step 1: Receive OneResult of the M
Step 2: Add this OneResult into the results of the sentiment classification of the TES;
Step 3: Return the results of the sentiment classification of the TES;

4.2 Creating the sentiment lexicons in English

The section includes three parts: 4.2.1, 4.2.2, and 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 RC through a Google search engine with AND operator and OR operator, as the following diagram in Fig. 4 below shows.

According to [14-18], 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

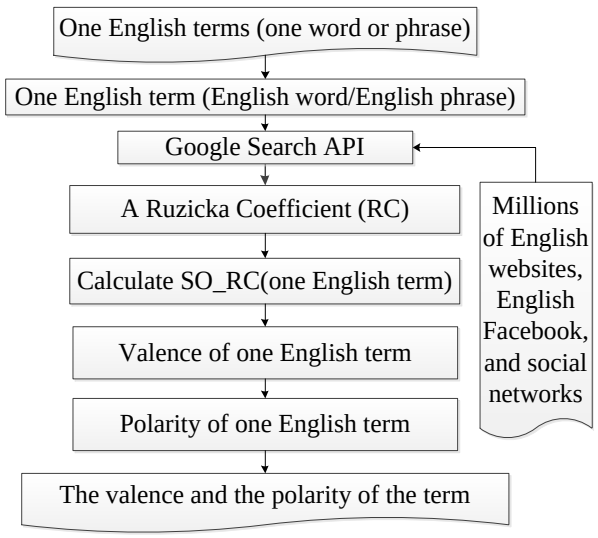

Figure. 4 Overview of identifying the valence and the polarity of one term in English using a RC

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

In [14-16] 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}. The AltaVista search engine is used in the PMI equations of [14-16]. [17] and [18] also use the PMI equations and Jaccard equations with the Google search engine in English.

According to [17-21], 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

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

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

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

In [17-21] 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 [17, 18]. The authors in [22] 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 [23] 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 [24] 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 [25] 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 [26] 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 [14-26], PMI, Jaccard, Cosine, Ochiai, Sorensen, Tanimoto, and RC are the similarity measures between two words, and they can perform the same functions and with the same characteristics; so RC is used in calculating the valence of the words. In addition, we prove that RC can be used in identifying the valence of the English word through the Google search with the AND operator and OR operator.

With the RC in [30-32], we have the equation of the RC as follows:

$$Ruzicka\ Coefficient\ (a, b) = Ruzicka\ Measure(a, b) = RC(a, b) = \frac{min[(a \cap b), (\neg a \cap b)]}{max[(a \cap b), (\neg a \cap b)]} \quad (6)$$

with a and b are the vectors.

From Eqs. (1), (2), (3), (4), (5), and (6), we propose many new equations of the RC 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.

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.

$$Ruzicka\ Measure(w1,w2) = Ruzicka\ Coefficient(w1,w2) = RC\ (w1,w2) = \frac{min[P(w1,w2),P(\neg w1,w2)]}{max[P(w1,w2),P(\neg w1,w2)]} \quad (7)$$

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

$$Valence(w) = SO\_RC(w) = RC(w,positive\_query) - RC(w,negative\_query) \quad (8)$$

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

$$RC\ (w,positive\_query\ ) = \frac{\min[P(w,positive_{query}),P(\neg w,positive_{query})]}{\max[P(w,positive_{query}),P(\neg w,positive_{query})]} \quad (9)$$

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

$$RC\ (w,negative\_query) = \frac{\min[P(w,negative_{query}),P(\neg w,negative_{query})]}{\max[P(w,negative_{query}),P(\neg w,negative_{query})]} \quad (10)$$

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_RC(w): valence of English word (or English phrase) w; is SO of word (or phrase) by using the RC. (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 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 RC (w, positive_query) is as equal as 1. The English word w is the farthest of positive_query if RC(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 RC(w, positive_query) > 0 and RC(w, positive_query) ≤ 1. The English word w is the nearest of negative_query if RC(w, negative_query) is as equal as 1. The English word w is the farthest of negative_query if RC(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 RC(w, negative_query) > 0 and RC(w, negative_query) ≤ 1. So, the valence of the English word w is the value of RC(w, positive_query) substrRCting the value of RC(w, negative_query) and the Eq. (8) is the equation of identifying the valence of the English word w.

We have RC as follows: (1)RC(w, positive_query) ≥ 0 and RC(w, positive_query) ≤ 1. (2)RC(w, negative_query) ≥ 0 and RC (w, negative_query) ≤ 1. (3)If RC(w, positive_query) = 0 and RC(w, negative_query) = 0 then SO_RC (w) = 0. (4)If RC(w, positive_query) = 1 and RC(w, negative_query) = 0 then SO_RC(w) = 0. (5)If RC(w, positive_query) = 0 and RC(w, negative_query) = 1 then SO_RC (w) = -1. (6)If RC(w, positive_query) = 1 and RC(w, negative_query) = 1 then SO_RC(w) = 0. So, SO_RC (w) ≥ -1 and SO_RC (w) ≤ 1.

The polarity of the English word w is positive polarity If SO_RC (w) > 0. The polarity of the English word w is negative polarity if SO_RC (w) < 0. The polarity of the English word w is neutral polarity if SO_RC (w) = 0. In addition, the semantic value of the English word w is SO_RC (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 Eqs. (8), (9), and (10), we identify the valence and the polarity of one word (or one phrase) in English by using the RC through the Google search engine with AND operator and OR operator.

4.2.2. Creating a bESD in a sequential environment

In this part, we calculate the valence and the polarity of the English words or phrases for our bESD by using the RC in a sequential system. We proposed the algorithm 25 to perform this section:
Input: the 55,000 English terms; the Google search engine
Output: a bESD
Step 1: Each term in the 55,000 terms, do repeat:
Step 2: By using Eqs. (8), (9), and (10) of the calculating a valence of one word (or one phrase) in English in the section (4.2.1), the sentiment score and the polarity of this term are identified. The valence and the polarity are calculated by using the RC through the Google search engine with AND operator and OR operator.
Step 3: Add this term into the bESD;
Step 4: End Repeat – End Step 1;
Step 5: Return bESD;

4.2.3. Creating a bESD in a distributed system

In this part, we identify the valence and the polarity of the English words or phrases for our bESD by using the RC in a DNE. This section includes two phases: the M phase and the R phase. The input of the M is the 55,000 terms in English in [27-29]. The output of the M is one term which the sentiment score and the polarity are identified. The output of the M is the input of the R. Thus, the input of the R is one term which the sentiment score and the polarity are identified. The output of the R is the bESD.

We built the algorithm 26 to implement the M:
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 Eqs. (8), (9), and (10) of the calculating a valence of one word (or one phrase) in English in the section 4.2.1, the sentiment score and the polarity of this term are identified. The valence and the polarity are calculated by using the RC through the Google search engine with AND operator and OR operator.
Step 3: Return this term;

We proposed the algorithm 30 to perform the R:
Input: one term which the sentiment score and the polarity are identified – The output of the M
Output: a bESD
Step 1: Add this term into the bESD;

Step 2: Return bESD;

5. Experiment

We have measured an Accuracy (A) to calculate the accuracy of the results of emotion classification. We used a Java programming language (JPL) for programming to save data sets, implementing our proposed model to classify the 7,500,000 documents of the TES and the 3,000,000 sentences of the TRA. To implement the proposed model, we have already used the JPL to save the TES and to save the results of emotion classification in English.

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

Our model related to the SOM, a TES and a TRA with the one-dimensional vectors and the RC 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 JPL is used in programming our model related to the SOM, a TES and a TRA with the one-dimensional vectors and the RC

The proposed model related to the SOM, a TES and a TRA with the one-dimensional vectors and the RC is performed in the CDNS with the configuration as follows: This CDNS includes 9 nodes (9 servers). The configuration of each server in the CDNS 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 JPL is used in programming the application of the proposed model related to the SOM, a TES and a TRA with the one-dimensional vectors and the RC in the Cloudera

In Table 1, the results of the documents of the English TES to test are presented.

Table 1. The results of the documents in the TES.

	TES	**Correct Classification**	**Incorrect Classification**
Negative	3,750,000	3,324,022	425,978
Positive	3,750,000	3,323,978	426,022
Summary	7,500,000	6,648,000	852,000

Table 2. The accuracy of our new model for the documents in the TES.

Proposed Model	Class	Accuracy
Our new model	Negative	88.64 %
	Positive	

Table 3. Average time of the classification of our new model for the documents in the TES.

	Average time of the classification / 7,500,000 documents.
This novel model in the sequential environment	32,537,841 seconds
This novel model in the Cloudera distributed system with 3 nodes	9,512,643 seconds
This novel model in the Cloudera distributed system with 6 nodes	5,586,312 seconds
This novel model in the Cloudera distributed system with 9 nodes	3,627,547 seconds

The accuracy of the sentiment classification of the documents in the TES in English is shown in Table 2 above.

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

Based on the above results in the tables, this work has seemed to be better than the researches related to the same topic. The above results have outperformed very well.

In Table 4 below, we compare this novel model with the research [33] as follows:

Table 4. Comparing this proposed model with the survey [33]

Name	**This model**	**The research[33]**
Big Data sentiment classification	Yes	Yes
English Sentiment Dictionary – sentiment lexicons	Yes	Yes
Training data set	Yes The sentences in English	No
Testing data set	Yes	Yes
Sequential environment	Yes	Yes
Distributed environment	Yes	Yes

Name	This model	The research[33]
One-dimensional vector	Yes	No
Multi-dimensional vector	No	No
Coefficient measure	Yes Ruzicka Coefficient (RC)	Yes JOHNSON Coefficient (JC)
Google search engine with AND operator and OR operator	Yes	Yes
Language	English	English
It can be applied to other languages	Yes	Yes
Accuracy	88.64 %	87.56%
Algorithms	-Self-Organizing Map Algorithm (SOM) of the machine learning -Ruzicka Coefficient (RC)	No Only JOHNSON Coefficient (JC)
Cloudera system	Yes	Yes
Hadoop Map and Hadoop Reduce	Yes	Yes
Nodes (servers)	9	9
vector space modeling (VSM)	No	No

6. Conclusion

In this survey, a new model has been proposed to classify sentiment of many documents in English using the SOM, a TES with the one-dimensional vectors and a RC with M/R in the CDNS. Based on our proposed new model, we have achieved 88.64% accuracy of the TES 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 TES in which the data sets are small in this survey.

According to Table 3, the average time of the sentiment classification of using the SOM, a TES and a TRA with the one-dimensional vectors and the RC in the sequential environment is 32,537,841 seconds / 7,500,000 English documents and it is greater than the average time of the sentiment classification of using the SOM, a TES and a TRA with the one-dimensional vectors and the RC in the CDNS with 3 nodes which is 9,512,643 seconds / 7,500,000 English documents. The average time of the sentiment classification of using the SOM, a TES and a TRA with the one-dimensional vectors and the RC in the CDNS with 9 nodes is 3,627,547 seconds / 7,500,000 English documents, and It is the shortest time in the table. Besides, the average time of the sentiment classification of using the SOM, a TES and a TRA with the one-dimensional vectors and the RC in the CDNS with 6 nodes is 5,586,312 seconds/7,500,000 English documents

The execution time of using the SOM, a TES and a TRA with the one-dimensional vectors and the RC in the CDNS is dependent on the performance of the CDNS and also dependent on the performance of each server on the Cloudera system.

The accuracy of the proposed model is dependent on many factors: (1)The SOM – related algorithms. (2)The TES. (3)The documents of the TES must be standardized carefully. (4)Transferring one sentence into one one-dimensional vector. (5)The RC – related algorithms

The execution time of the proposed model is dependent on many factors: (1)The parallel network environment such as the CDNS. (2)The distributed functions such as M and R. (3)The performance of the DNE. (4)The number of nodes of the PNE. (5)The performance of each node (each server) of the DNE. (6)The sizes of the TRA and the TES. (7)Transferring one sentence into one one-dimensional vector.

The proposed model has many advantages and disadvantages. Its positives are as follows: It uses the SOM, a TES and a TRA with the one-dimensional vectors and the RC 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.

Acknowledgments

This research is funded by Nguyen Tat Thanh University, 300A Nguyen Tat Thanh Street, Ward 13, District 4, Ho Chi Minh City, 702000, Vietnam.

References

[1] T. Kohonen, "The self-organizing map", In: *Proc. of the IEEE*, Vol. 78, No. 9, pp. 1464 - 1480, 1990.

[2] J.L. Giraudel and S. Lek, "A comparison of self-organizing map algorithm and some conventional statistical methods for ecological community ordination", *International Journal of Ecological Modelling*, Vol. 146, No. 1–3, pp.329-339, 2001.

[3] J. Vesantom and E. Alhoniemi, "Clustering of the self-organizing map", *IEEE Transactions on Neural Networks*, Vol. 11, No. 3, pp. 586-600, 2000.

[4] V. Kant Singh, and V. Kumar Singh, "Vector Space Model: An Information Retrieval System", *Int. J. Adv. Engg. Res.* Studies/IV/II/Jan.-March, Vol. 87, No. IV/II, pp. 141-143,2015.

[5] V. Carrera-Trejo, G. Sidorov, S. Miranda-Jiménez, M. Moreno Ibarra, and R. 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, pp. 7-19, 2015.

[6] P. Soucy and G. W. Mineau, "Beyond TFIDF Weighting for Text Categorization in the Vector Space Model", In: *Proc. of the 19th International Joint Conference on Artificial Intelligence*, pp. 1130-113, 2015.

[7] P. Vo Ngoc and T. Phan Thi, "Sentiment classification using Enhanced Contextual Valence Shifters", In: *Proc. of International Conf. on Asian Language Processing*, pp. 224-229, 2014.

[8] T. Vo Thi Ngoc, P. Vo Ngoc, and T. Phan Thi, "Learning More Chi Square Feature Selection to Improve the Fastest and Most Accurate Sentiment Classification", In: *Proc. of The Third Asian Conference on Information Systems*, pp. 1-10, 2014

[9] D. Nguyen Duy, P. Vo Ngoc, T. Vo Thi Ngoc, and C. Vo Thi Ngoc, "STING Algorithm used English Sentiment Classification in A Parallel Environment", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 31, No. 7, pp. 1-31, 2017.

[10] P. Vo Ngoc, D. Nguyen Duy, C. Vo Thi Ngoc, and T. Vo Thi Ngoc, "Fuzzy C-Means for English Sentiment Classification in a Distributed System", *International Journal of Applied Intelligence*, 2016.

[11] P. Vo Ngoc, Chau Vo Thi Ngoc, Tran Vo THi Ngoc, and D. Nguyen Duy, "A C4.5 algorithm for english emotional classification", *International Journal of Evolving Systems*, pp.1-27, 2017.

[12] P. Vo Ngoc, C. Vo Thi Ngoc, and T. Vo Thi Ngoc, "SVM for English Semantic Classification in Parallel Environment", *International Journal of Speech Technology*, pp.1-30, 2017.

[13] P. Vo Ngoc, T. Vo Thi Ngoc, C. Vo Thi Ngoc, D. Nguyen Duy, and K. Ly Doan Duy, "A Decision Tree using ID3 Algorithm for English Semantic Analysis", *International Journal of Speech Technology*, pp.1-22, 2017.

[14] B. Aleksander and H. Hammer, "Constructing sentiment lexicons in Norwegian from a large text corpus", In: *Proc. of 2014 IEEE 17th International Conference on Computational Science and Engineering*, 2014.

[15] P.D. Turney and M.L. Littman, "Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus", *arXiv:cs/0212012, Learning (cs.LG); Information Retrieval (cs.IR)*, pp.1-11, 2002.

[16] R. Malouf and T. Mullen, "Graph-based user classification for informal online political discourse", In: *Proc. of the 1st Workshop on Information Credibility on the Web*, pp.1-8, 2017.

[17] S. Feng, L. Zhang, B. Li Daling Wang, G. Yu, and W. Kam-Fai, "Is Twitter A Better Corpus for Measuring Sentiment Similarity?", In: *Proc. of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 897–902, 2013.

[18] A. Nguyen Thi Thu and H. Masafumi, "Adjective-Based Estimation of Short Sentence's Impression", In: *Proc. of the 5th Kanesi Engineering and Emotion Research*, pp. 1219-1234, 2014.

[19] N. R. Shikalgar and A. M. Dixit, "JIBCA: Jaccard Index based Clustering Algorithm for Mining Online Review", *International Journal of Computer Applications*, Vol. 105, No. 15, pp. 23-28, 2014.

[20] X. Ji, C. Soon Ae, W. Zhi, and J. Geller, "Twitter sentiment classification for measuring public health concerns", *Soc. Netw. Anal. Min*, Vol. 5, No. 13, pp. 1-25, 2015.

[21] N. Omar, M. Albared, A. Qasem Al-Shabi, and T. Al-Moslmi, "Ensemble of Classification algorithms for Subjectivity and Sentiment Analysis of Arabic Customers' Reviews", *International Journal of Advancements in Computing Technology*, Vol. 5, No. 14, pp. 77-85, 2013.

[22] P. Vo Ngoc, C. Vo Thi Ngoc, T. Vo Thi Ngoc, and D. Nguyen Duy, "A Vietnamese adjective emotion dictionary based on exploitation of Vietnamese language characteristic", *International Journal of Artificial Intelligence Review*, Vol. 50, No. 1, pp. 93–159, 2018.

[23] P. Vo Ngoc, C. Vo Thi Ngoc, D. Nguyen Duy, T. Vo Thi Ngoc, and Tuan A. Nguyen, "A Valences-Totaling Model for English Sentiment Classification", *International Journal of Knowledge and Information Systems*, Vol. 53, No. 3, pp. 579–636, 2017.

[24] P. Vo Ngoc, C. Vo Thi Ngoc, and T. Vo Thi Ngoc, "Shifting Semantic Values of English Phrases for Classification", *International Journal of Speech Technology*, Vol. 20, No. 3, pp. 509–533, 2017.

[25] P. Vo Ngoc, C. Vo Thi Ngoc, T. Vo Thi Ngoc, Nguy Duy Dat, and K. Ly Doan Duy, "A Valence-Totaling Model for Vietnamese Sentiment Classification", *International Journal of Evolving Systems*, pp. 1-47, 2017.

[26] P. Vo Ngoc, C. Vo Thi Ngoc, T. Vo Thi Ngoc, D. Nguyen Duy, and K. Ly Doan Duy, "Semantic Lexicons of English Nouns for Classification", *International Journal of Evolving Systems*, pp. 1-65, 2017.

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

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

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

[30] C. Seung-Seok, C. Sung-Hyuk, and C. C. Tappert, "A Survey Of Binary Similarity And Distance Measures", *Systemics, Cybernetics And Informatics*, Vol. 8, No. 1, 2010.

[31] J. Torres-Sospedra, R. Montoliu, S. Trilles, O. Belmonte, and J. Huerta, "Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems", *Expert Systems with Applications*, Vol. 42, No.23, pp. 9263-9278, 2015.

[32] R. E. Tulloss, *Assessment of Similarity Indices for Undesirable Properties and a new Tripartite Similarity Index Based on Cost Functions*. Offprint from Palm, M. E. and I. H. Chapela, eds. 1997. MRCology in Sustainable Development: Expanding Concepts, Vanishing Borders. (Parkway Publishers, Boone, North Carolina), pp. 122-143, 1997

[33] P. Vo Ngoc and T. Vo Thi Ngoc, "English Sentiment Classification using Only the Sentiment Lexicons with a JOHNSON Coefficient in a Parallel Network Environment", *American Journal of Engineering and Applied Sciences*, Vol. 11, No. 1, pp. 38-65, 2018

Abbreviation

Ruzicka Coefficient: RC
A Self-Organizing Map Algorithm: SOM
vector space modeling: VSM
basis English sentiment dictionary: bESD
Cloudera distributed network system: CDNS
Hadoop Map: M
Hadoop Reduce: R
sequential system: SS
sequential environment: SE
distributed network environment: DNE
parallel network environment: PNE
Pointwise Mutual Information: PMI
Java programming language: JPL
testing data set: TES
training data set: TRA