THE BINARY BITS AND THE SENTIMENT LEXICONS BASED ON AN YULE'S SIGMA COEFFICIENT USED FOR SENTIMENT CLASSIFICATION IN ENGLISH

[1]DR.VO NGOC PHU, [2]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 already surveyed many different approaches for sentiment classification to get higher accuracies, to shorten execution times, and to save a lot of storage spaces for many years because many significant contributions of the sentiment classification have widely been applied to everyday life, political activities, commodity production, and commercial activities. As known, many binary bits can help us save a lot of storage spaces. Therefore, we have proposed a novel model in this survey using the binary bits and many sentiment lexicons according to an Yule's Sigma coefficient (YSC) to classify 9,000,000 documents of our teting data set comprising the 4,500,000 positive and the 4,500,000 negative into either the positive polarity or the negative polarity based on 5,000,000 sentences of our training data set including the 2,500,000 positive and the 2,500,000 negative in English. We do not use any vector space modeling (VSM). In addition, we do not use any one-dimensional vectors. We also do not use any multi-dimensional vectors. We only use the binary bits and the sentiment values of the lexicons for the proposed model. We use many similarity measures of the YSC. One sentence is transferred into the binary bits (the binary bits string) according to the sentiment scores of the lexicons. All the positive sentences of training data set are transferred into all the positive binary bits strings of training data set, called the positive group and all the negative sentences of training data set are also transferred into all the negative binary bits strings of training data set, called the negative group. Based on many similarity measures by using the YSC, one sentence is clustered into either the positive group or the negative group. The sentiment classification of one document of the testing data set is identified according to the sentiment classification results of all the sentences of this document. The novel model has firstly been implemented in a sequential environment. Next, we have performed the proposed model in a parallel network environment secondly. The execution time of the distributed network system is faster than that of the sequential environment. There has been an accuracy 89.01% of the testing data set in this survey. The sentiment classification of many millions of the documents can be identified by using this novel model successfully. Many advantage results of our novel model can widely be applied to many different fields in many commercial applications and surveys of the sentiment classification.

Keywords: *English sentiment classification; parallel system; Cloudera; Hadoop Map and Hadoop Reduce; binary bits; sentiment scores of lexicons; Yule's Sigma coefficient.*

1. INTRODUCTION

We have already surveyed many different approaches for sentiment classification to get higher accuracies, to shorten execution times, and to save a lot of storage spaces for many years because many significant contributions of the sentiment classification have widely been applied to everyday life, political activities, commodity production, and commercial activities. Many different approaches related sentiment lexicons have already been proposed for sentiment classification. There are the significant approaches related to the sentiment lexicons in [1-32].

A set of objects is processed into classes of similar objects, call clustering data in many clustering technologies of a data mining field. A set of data objects are similar to each other, called one cluster and the data objects 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.

Binary bit strings are arbitrary sequences of zero or more binary digits (bits), each having a value of 0 or 1 and a length of 1 bit. A binary code represents text, computer processor instructions, or other data using any two-symbol system, but often the binary number system's 0 and 1. The binary code assigns a pattern of binary digits (bits) to each character, instruction, etc. For example, a binary string of eight bits can represent any of 256 possible values and can therefore represent a variety of different items. In computing and telecommunications, binary codes are used for various methods of encoding data, such as character strings, into bit strings.

The motivation of this new model is as follows: We always want to find many novel approaches for the sentiment lexicons with many advantages of the results such as a high accuracy, shortening a execution time, a low cost, no depending on any training data sets, etc. As known, many binary bits can help us save a lot of storage spaces. Therefore, we survey the binary bits and the sentiment values of the lexicons to classify one document into either the positive polarity or the negative polarity. Besides, we can implement the binary bits and the sentiment lexicons in both a sequential system and a distributed network environment. This will result in many discoveries in scientific research, hence the motivation for this study.

Our proposed model has its novelty and originality as follows:

1)The sentiment scores of the lexicons were applied to the sentiment classification for one document in English.

2)This can also be applied to identify the sentiments (positive, negative, or neutral) of millions of many documents.

3)This survey can be applied to other parallel network systems.

4)The Cloudera, Hadoop Map (M) and Hadoop Reduce (R) were used in the proposed model.

5)There was not any vector space modeling (VSM) in this survey.

6)We did not use any one-dimensional vectors

7)We did not use any multi-dimensional vectors.

8)We used many sentiment lexicons.

9)We used many binary bits strings.

10)The input of the novel model is the documents of the testing data set and the sentences of the training data set. We studied to transfer them into the formats according to the valences of the lexicons and the binary bits strings to classify the documents of the testing data set into either the positive polarity or the negative polarity.

11)The novel model was implemented in both a sequential environment and a parallel network system.

12)We built the algorithms related to the sentiment lexicons for the novel model.

13)The algorithms related to Hadoop Map and Hadoop Reduce in the Cloudera system were proposed certainly in this study.

14)The sentiment lexicons – related algorithms were proposed in the distributed environment.

15)We used the Yule's Sigma coefficient (YSC) to identify many sentiment values and polarities of the sentiment lexicons of our basis English sentiment dictionary (bESD) through a Google search engine with AND operator and OR operator.

16)In this survey, we only used the binary bits and the sentiment lexicons for the sentiment classification.

17)The sentiment lexicons were used in transferring one sentence into one binary bits string in both a sequential system and a distributed network environment.

18)The binary bits – related algorithms were built in this research.

Therefore, we have studied this model in more details.

With the purpose of this survey, we always try to find a new approach to reform the accuracy of the sentiment classification results and to shorten the execution time of the proposed model with a low cost. We also try to find a new approach to save a lot of storage spaces of many big data sets and the results of the sentiment classification.

We want to get higher accuracy, shorten execution time of the sentiment classification and to save a lot of storage spaces. Therefore, there is not any vector space modeling (VSM) in this survey. We do not use any one-dimensional vectors. We also do not use any multi-dimensional vectors. We only use the binary bits and the sentiment lexicons. We use the YSC to calculate the sentiment values and polarities of the sentiment lexicons of the bESD through the Google search engine with AND operator and OR operator. Based on the sentiment lexicons of the bESD, we transfer one sentence into one binary bits string. We use many similarity measures of the YSC to cluster one sentence into either the positive polarity or the negative polarity.

The novel model is performed in this survey as follows: we firlsty identify the sentiment scores and polarities of the sentiment lexicons of the bESD by using the YSC through the Google search engine with AND operator and OR operator. According to the surveys related to the binary code of letters in English in [47-52] and the researches related to transferring a decimal to a binary code in [53-58], we transfer one sentence into one binary bits string. One binary bits string is one binary bits vector with each element of this vector is a bit 0 or 1. A positive group is a group which we transfer all the positive sentences of the training data set into the positive binary bits strings and a negative group is a group which we also transfer all the negative sentences of the training data set into the negative binary bits strings. All the sentences of one document of the testing data set are transferred into the binary bits strings. Each binary bits string in the binary bits strings of this document, we calculate a total of all the similarity measures of this string with all the positive binary bits string of the positive group by using the YSC, called ATotalOfPositiveSimilarityMeasures, and we also calculate a total of all the similarity measures of this string with all the negative binary bits string of the negative group by using the YSC, called ATotalOfNagativeSimilarityMeasures. This binary bits string is clustered into the positive if ATotalOfPositiveSimilarityMeasures is greater than ATotalOfNegativeSimilarityMeasures. This binary bits string is clustered into the negative if ATotalOfPositiveSimilarityMeasures is less than ATotalOfNegativeSimilarityMeasures. This binary bits string is clustered into the neutral if ATotalOfPositiveSimilarityMeasures is as equal as ATotalOfNegativeSimilarityMeasures. One document is clustered into the positive if the number of the binary bits strings clustered into the positive is greater than that clustered into the negative in this document. The document is clustered into the negative if the number of the binary bits strings clustered into the positive is less than that clustered into the negative in this document. One document is clustered into the neutral if the number of the binary bits strings clustered into the positive is as equal as that clustered into the negative in this document. Finally, the sentiment classification of all the documents of the testing data set is identified certainly.

We firstly perform all the above things in the sequential 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. Then, we secondly implement all the above things 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 novel model.

This novel 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)Many crucial contributions are listed in the Future Work section.
6)The algorithm of data mining is applicable to semantic analysis of natural language processing.
7)This study also proves that different fields of scientific research can be related in many ways.
8)Millions of English documents are successfully processed for emotional analysis.
9)The semantic classification is implemented in the parallel network environment.
10)The principles are proposed in the research.
11)The Cloudera distributed environment is used in this study.
12)The proposed work can be applied to other distributed systems.
13)This survey uses Hadoop Map (M) and Hadoop Reduce (R).
14)Our proposed model can be applied to many different parallel network environments such as a Cloudera system
15)This study can be applied to many different distributed functions such as Hadoop Map (M) and Hadoop Reduce (R).
16)The YSC – related algorithms are built in this study.
17)The binary bits – related algorithms are proposed in this work.

This study contains 6 sections. Section 1 introduces the study; Section 2 discusses the related works about the binary bits, Yule's Sigma coefficient (YSC), 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. 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'S SIGMA coefficient (YSC) 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 the similarity coefficients in [1-27]. In the research [1], 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. The methodology was based on the Point wise Mutual Information (PMI). The authors introduced a modification of the PMI that considers small "blocks" of the text instead of the text as a whole. The study in [2] introduced a simple algorithm for unsupervised learning of semantic orientation from extremely large corpora, etc.

The surveys related to the similarity coefficients to calculate the valences of words are in [28-32].

The English dictionaries are [33-38] and there are more than 54,000 English words (including English nouns, English adjectives, English verbs, etc.) from them.

The research projects related to implementing algorithms, applications, studies in parallel network environment in [39-41]. In [39, 40], Hadoop is an Apache-based framework used to handle large data sets on clusters consisting of multiple computers, using the Map and Reduce programming model. The two main projects of the Hadoop are Hadoop Distributed File System (HDFS) and Hadoop M/R (Hadoop Map /Reduce). Hadoop M/R allows engineers to program for writing applications for parallel processing of large data sets on clusters consisting of multiple computers. A M/R task has two main components: (1) Map and (2) Reduce. This framework splits inputting data into chunks which multiple Map tasks can handle with a separate data partition in parallel. The outputs of the map tasks are gathered and processed by the Reduce task ordered. The input and output of each M/R task are stored in HDFS because the Map tasks and the Reduce tasks perform on the pair (key, value), and formatted input and output formats will be the pair (key, value). Cloudera [41], the global provider of the fastest, easiest, and most secure data management and analytics platform built on Apache™ Hadoop® and the latest open source technologies, announced today that it will submit proposals for Impala and Kudu to join the Apache Software Foundation (ASF). By donating its leading analytic database and columnar storage projects to the ASF, Cloudera aims to accelerate the growth and diversity of their respective developer communities. Cloudera delivers the modern data management and analytics platform built on Apache Hadoop and the latest open source technologies. The world's leading organizations trust Cloudera to help solve their most challenging business problems with Cloudera Enterprise, the fastest, easiest and most secure data platform available to the modern world. Cloudera's customers efficiently capture, store, process, and analyze vast amounts of data, empowering them to use advanced analytics to drive business decisions quickly, flexibly, and at lower cost than has been possible before. To ensure Cloudera's customers are successful, it offers comprehensive support, training and professional services.

There are the works related to the Yule's Sigma coefficient (YSC) in [42-46]. The binary similarity and dissimilarity measures in [42] had critical roles in the processing of data consisting of binary vectors in various fields including bioinformatics and chemometrics. These metrics expressed the similarity and dissimilarity values between two binary vectors in terms of the positive matches, absence mismatches or negative matches. The authors in [43] collected 76 binary similarity and distance measures used over the last century and reveal their correlations through the hierarchical clustering technique. In [44], an overview of ARMA spectral estimation techniques based on the modified Yule-Walker equations was presented. The importance of using order overestimation, as well as of using an overdetermined set of equations, was emphasized, etc.

The surveys related to the binary code of letters in English are shown in [47-52]. The researches in [47-52] showed all the binary codes of all the letters in English completely.

There are the researches related to transferring a decimal to a binary code in [53-58]. The surveys in [53-58] showed how to transfer one decimal to one binary code.

The latest researches of the sentiment classification are [59-69]. In the research [59], 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. The survey in [60] discussed 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 presented opinion detection and organization subsystem, which have already been integrated into our larger question-answering system, etc.

3. DATA SET

In Fig 1, we built our testing data set including 9,000,000 documents in the movie field, which contains 4,500,000 positive documents and 4,500,000 negative documents in English. All English 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.

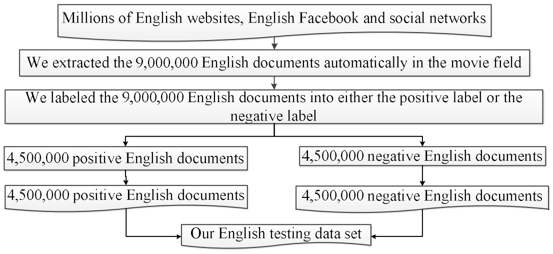
Millions of English websites, English Facebook and social networks
We extracted the 9,000,000 English documents automatically in the movie field
We labeled the 9,000,000 English documents into either the positive label or the negative label
4,500,000 positive English documents
4,500,000 negative English documents
4,500,000 positive English documents
4,500,000 negative English documents
Our English testing data set

Fig. 1: Our English Testing Data Set.

In Fig 2, we built our testing data set comprising 5,000,000 sentences in the movie field, which contains 2,500,000 positive sentences and 2,500,000 negative sentences in English. All the English sentences 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.

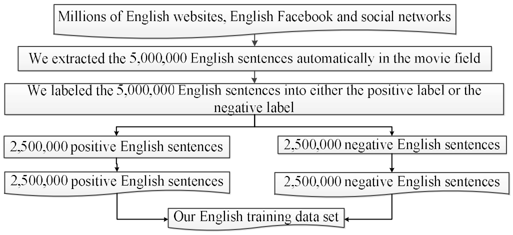
Millions of English websites, English Facebook and social networks
We extracted the 5,000,000 English sentences automatically in the movie field
We labeled the 5,000,000 English sentences into either the positive label or the negative label
2,500,000 positive English sentences
2,500,000 negative English sentences
2,500,000 positive English sentences
2,500,000 negative English sentences
Our English training data set

Fig. 2: Our English Training Data Set.

4. METHODOLOGY

This section comprises three parts as follows: (4.1); (4.3); and (4.2). The first part is the sub-section (4.1) which we create the sentiment lexicons of the bESD in both a sequential environment and a distributed network system. The second part is the sub-section (4.2) which we use the sentiment lexicons to transfer all the sentences of one document of the testing data set and all the sentences of the training data set into the binary bits strings in both a sequential system and a parallel network environment. The third part is the sub-section (4.3) which we use the binary bits and sentiment lexicons to classify the sentiments (positive, negative, or neutral) for the documents of the testing data set in both a sequential environment and a distributed system.

The sub-section (4.1) comprises three sub-sections as follows: (4.1.1); (4.1.2); and (4.1.3). In the sub-section (4.1.1), we calculate a valence of one term (meaningful word or meaningful phrase). In the sub-section (4.1.2), we create the sentiment lexicons of the bESD in a sequential environment. In the sub-section (4.1.3), we create the sentiment lexicons of the bESD in a distributed network system.

The sub-section (4.2) includes threes sub-section as follows: (4.2.1); (4.2.2); and (4.2.3). In the sub-section (4.2.1), we display the theories of transferring one sentence into one binary bits string. In the sub-section (4.2.2), we use the sentiment lexicons to transfer all the sentences of one document of the testing data set and all the sentences of the training data set into the binary bits strings in a sequential system. In the sub-section (4.2.3), we use the sentiment lexicons to transfer all the sentences of one document of the testing data set and all the sentences of the training data set into the binary bits strings in a parallel network environment

The sub-section (4.3) has two sub-sections as follows: (4.3.1) and (4.3.2). In the sub-section (4.3.1), we use the binary bits and sentiment lexicons to classify the sentiments (positive, negative, or neutral) for the documents of the testing data set in a sequential environment. In the sub-section (4.3.2), we use the binary bits and sentiment lexicons to classify the sentiments (positive, negative, or neutral) for the documents of the testing data set in a distributed system.

4.1 The sentiment lexicons of our basis English sentiment dictionary (bESD)

This section comprises three sub-sections as follows: (4.1.1); (4.1.2); and (4.1.3). In the sub-section (4.1.1), we calculate a valence of one term (meaningful word or meaningful phrase). In the sub-section (4.1.2), we create the sentiment lexicons of the bESD in a sequential environment. In the sub-section (4.1.3), we create the sentiment lexicons of the bESD in a distributed network system.

4.1.1 Calculating the valence of one sentiment lexicon

In this part, we calculate the valence and the polarity of one English word (or phrase) by using the YSC through a Google search engine with AND operator and OR operator, as the following diagram in Fig 3 below shows.

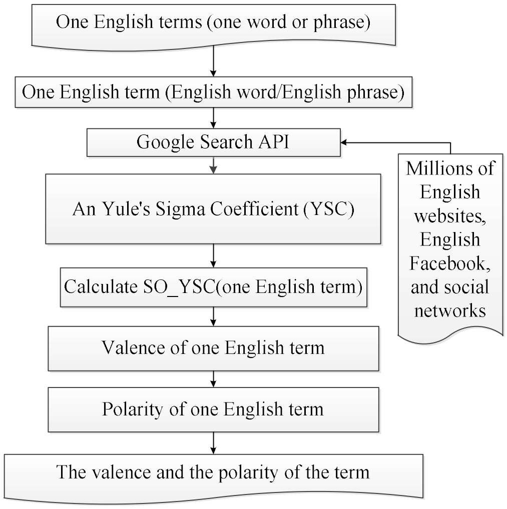
One English terms (one word or phrase)
One English term (English word/English phrase)
Google Search API
Millions of English websites, English Facebook, and social networks
An Yule's Sigma Coefficient (YSC)
Calculate SO_YSC(one English term)
Valence of one English term
Polarity of one English term
The valence and the polarity of the term

Fig. 3: Overview Of Identifying The Valence And The Polarity Of One Term In English Using An Yule's Sigma Coefficient (YSC)

According to [1-15], Pointwise Mutual Information (PMI) between two words wi and wj has the equation

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

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

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

In [1-8] 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 [2, 3, 5] and the Google search engine is used in the PMI equations of [4, 6, 8]. Besides, [4] also uses German, [5] also uses Macedonian, [6] also uses Arabic, [7] also uses Chinese, and [8] also uses Spanish. In addition, the Bing search engine is also used in [6]. With [9-12], the PMI equations are used in Chinesese, not English, and Tibetan is also added in [9]. About the search engine, the AltaVista search engine is used in [11] and [12] 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 [13]. [14] and [15] also use the PMI equations and Jaccard equations with the Google search engine in English. In [14-21] 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 [14, 15, 17]. [16] and [21] use the Jaccard equations in English. [20] and [22] use the Jaccard equations in Chinese. [18] uses the Jaccard equations in Arabic. The Jaccard equations with the Chinese search engine in Chinese are used in [19].The authors in [28] 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 [29] 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 [30] 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 [31] 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 [32] 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 Vietnamse. The Ochiai is used with the Google in Vietnamese. The Cosine and Sorensen are used with the Google in English.

According to [1-32], PMI, Jaccard, Cosine, Ochiai, Sorensen, Tanimoto and Yule's Sigma coefficient (YSC) are the similarity measures between two words, and they can perform the same functions and with the same characteristics; so YSC is used in calculating the valence of the words. In addition, we prove that RTC 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's Sigma coefficient (YSC) in [45-49], we have the equation of the YSC as follows:

$$\text{Yule's Sigma coefficient}(a,b) = \text{Yule's Sigma Measure}(a,b) = \mathrm{YSC}(a,b) = \frac{A3}{B3} \qquad (3)$$

with a and b are the vectors.

$$A3 = \sqrt{(a \cap b) * (\neg a \cap \neg b)} - \sqrt{(\neg a \cap b) * (a \cap \neg b)}$$

$$B3 = \sqrt{(a \cap b) * (\neg a \cap \neg b)} + \sqrt{(\neg a \cap b) * (a \cap \neg b)}$$

From the eq. (1), (2), (3), we propose many new equations of the YSC 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. (3), when a has only one element, a is a word. When b has only one element, b is a word. In eq. (3), a is replaced by w1 and b is replaced by w2.

$$\text{Yule's Sigma Measure}(w1,w2) = \text{Yule's Sigma Coefficient}(w1,w2) = \mathrm{YSC}\ (w1,w2) = \frac{A4}{B4} \qquad (4)$$

with $A4 = \sqrt{P(w1,w2) * P(\neg w1,\neg w2)} - \sqrt{P(\neg w1,w2) * P(w1,\neg\, w2)}$

$B4 = \sqrt{P(w1,w2) * P(\neg w1,\neg w2)} + \sqrt{P(\neg w1,w2) * P(w1,\neg\, w2)}$

Eq. (4) is similar to eq. (1). In eq. (2), eq. (1) is replaced by eq. (4). We have eq. (5) as follows:

$$\mathrm{Valence}(w) = \mathrm{SO\_YSC}(w) = \mathrm{YSC}(w, \mathrm{positive\_query}) - \mathrm{YSC}(w, \mathrm{negative\_query}) \qquad (5)$$

In eq. (5), w1 is replaced by w and w2 is replaced by position_query. We have eq. (6). Eq. (6) is as follows:

$$\mathrm{YSC}(w, \mathrm{positive\_query}) = \frac{A6}{B6} \quad (6)$$

with $A6 = \sqrt{P(w, \mathrm{positive\_query}) * P(\neg w, \neg \mathrm{positive\_query})} - \sqrt{P(\neg w, \mathrm{positive\_query}) * P(w, \neg\ \mathrm{positive\_query})}$

$B6 = \sqrt{P(w, \mathrm{positive\_query}) * P(\neg w, \neg \mathrm{positive\_query})} + \sqrt{P(\neg w, \mathrm{positive\_query}) * P(w, \neg\ \mathrm{positive\_query})}$

In eq. (4), w1 is replaced by w and w2 is replaced by negative_query. We have eq. (7). Eq. (7) is as follows:

$$\mathrm{YSC}(w, \mathrm{negative\_query}) = \frac{A7}{B7} \quad (7)$$

with $A7 = \sqrt{P(w, \mathrm{negative\_query}) * P(\neg w, \neg \mathrm{negative\_query})} - \sqrt{P(\neg w, \mathrm{negative\_query}) * P(w, \neg\ \mathrm{negative\_query})}$

$B7 = \sqrt{P(w, \mathrm{negative\_query}) * P(\neg w, \neg \mathrm{negative\_query})} + \sqrt{P(\neg w, \mathrm{negative\_query}) * P(w, \neg\ \mathrm{negative\_query})}$

We have the information about w, w1, w2, and etc. as follows:

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_YSC(w): valence of English word (or English phrase) w; is SO of word (or phrase) by using Yule's Sigma coefficient (YSC)

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,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).

14)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))).

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 YSC (w, positive_query) is as equal as 1. The English word w is the farthest of positive_query if YSC(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 YSC(w, positive_query) > 0 and YSC(w, positive_query) ≤ 1. The English word w is the nearest of negative_query if YSC(w, negative_query) is as equal as 1. The English word w is the farthest of negative_query if YSC(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 YSC(w, negative_query) > 0 and YSC(w, negative_query) ≤ 1. So, the valence of the English word w is the value of YSC(w, positive_query) substracting the value of YSC(w, negative_query) and the eq. (5), eq. (6) and eq. (7) is the equation of identifying the valence of the English word w.

We have the information about YSC as follows:

1)YSC(w, positive_query) ≥ 0 and YSC(w, positive_query) ≤ 1.

2)YSC(w, negative_query) ≥ 0 and YSC(w, negative_query) ≤ 1

3)If YSC(w, positive_query) = 0 and YSC(w, negative_query) = 0 then SO_YSC(w) = 0.

4)If YSC(w, positive_query) = 1 and YSC(w, negative_query) = 0 then SO_YSC(w) = 0.

5)If YSC(w, positive_query) = 0 and YSC(w, negative_query) = 1 then SO_YSC(w) = -1.

6)If YSC(w, positive_query) = 1 and YSC(w, negative_query) = 1 then SO_YSC(w) = 0.

So, SO_YSC(w) ≥ -1 and SO_YSC(w) ≤ 1.

The polarity of the English word w is positive polarity If SO_YSC(w) > 0. The polarity of the English word w is negative polarity if SO_YSC(w) < 0. The polarity of the English word w is neutral polarity if SO_YSC(w) = 0. In addition, the semantic value of the English word w is SO_YSC(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.

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

To understand many advantages of our novel model's results, we compare the results of the proposed model with the researches in the tables as follows: Table 1, Table 2, Table 8, and Table 9.

In Table 1, we present Comparisons of our model's results with the works related to [1-32].

The comparisons of our model's advantages and disadvantages with the works related to [1-32] are displayed in Table 2.

In Table 8, we show the comparisons of our model's results with the works related to the Yule's Sigma coefficient (YSC) in [45-49].

The comparisons of our model's benefits and drawbacks with the studies related to the Yule's Sigma coefficient (YSC) in [45-49] are presented in Table 9.

4.1.2 The sentiment lexicons of the bESD in the sequential environment

According to [33-38], 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 YSC in a sequential system, as the following diagram in Fig 4 below shows.

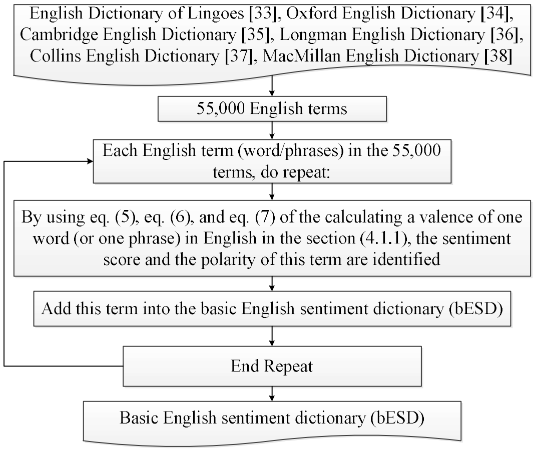
English Dictionary of Lingoes [33], Oxford English Dictionary [34], Cambridge English Dictionary [35], Longman English Dictionary [36], Collins English Dictionary [37], MacMillan English Dictionary [38]
55,000 English terms
Each English term (word/phrases) in the 55,000 terms, do repeat:
By using eq. (5), eq. (6), and eq. (7) 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
Add this term into the basic English sentiment dictionary (bESD)
End Repeat
Basic English sentiment dictionary (bESD)

Fig. 4: Overview Of Creating A Basis English Sentiment Dictionary (Besd) In A Sequential Environment

We proposed the algorithm 1 to perform this section.

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. (5), eq. (6) and eq. (7) 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 YSC 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 bESD has more 55,000 English words (or English phrases) and bESD is stored in Microsoft SQL Server 2008 R2.

4.1.3 The sentiment lexicons of the bESD in the distributed system

According to [33-38], 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 YSC in a parallel network environment, as the following diagram in Fig 5 below shows.

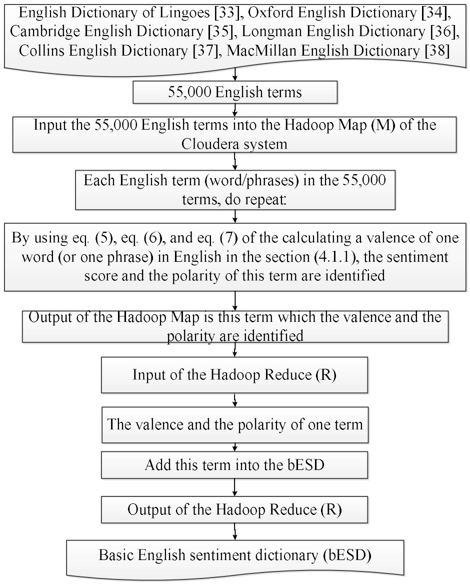
English Dictionary of Lingoes [33], Oxford English Dictionary [34], Cambridge English Dictionary [35], Longman English Dictionary [36], Collins English Dictionary [37], MacMillan English Dictionary [38]
55,000 English terms
Input the 55,000 English terms into the Hadoop Map (M) of the Cloudera system
Each English term (word/phrases) in the 55,000 terms, do repeat:
By using eq. (5), eq. (6), and eq. (7) 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
Output of the Hadoop Map is this term which the valence and the polarity are identified
Input of the Hadoop Reduce (R)
The valence and the polarity of one term
Add this term into the bESD
Output of the Hadoop Reduce (R)
Basic English sentiment dictionary (bESD)

Fig. 5: Overview Of Creating A Basis English Sentiment Dictionary (Besd) In A Distributed Environment

This section includes two phases as follows: 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 [33-38]. 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 2 to implement the Hadoop Map phase.

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

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

Step 0: Input the 55,000 English terms in the Hadoop Map in the Cloudera.
Step 1: Each term in the 55,000 terms, do repeat:
Step 2: By using eq. (5), eq. (6), and eq. (7) 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 YSC through the Google search engine with AND operator and OR operator.
Step 3: Return this term;

We built the algorithm 3 to perform the Hadoop Reduce pase.

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

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

4.2 Using the sentiment lexicons to transfer all the sentences of one document of the testing data set and all the sentences of the training data set into the binary bits strings in both a sequential system and a parallel network environment

This section includes threes sub-section as follows: (4.2.1); (4.2.2); and (4.2.3). In the sub-section (4.2.1), we display the theories of transferring one sentence into one binary bits string. In the sub-section (4.2.2), we use the sentiment lexicons to transfer all the sentences of one document of the testing data set and all the sentences of the training data set into the binary bits strings in a sequential system. In the sub-section (4.2.3), we use the sentiment lexicons to transfer all the sentences of one document of the testing data set and all the sentences of the training data set into the binary bits strings in a parallel network environment

4.2.1 Displaying the theories of transferring one sentence into one binary bits string

We encrypt the sentiment lexicons of the bESD to the bit arrays and each bit array in the bit arrays presents each term in the sentiment lexicons with the information as follows: a content of this term, a sentiment score of this term. This is called the bit arrays of the bESD which are stored in a small storage space.

We assume that the sentiment lexicons of the bESD is stored in the table as follows:

Ordering number	Lexicons	Valence
1	Good	+1
2	Very good	+2
3	Bad	-1
4	Very bad	-2
5	Terrible	-1.2
6	Very terrible	-2.3
…	…	…
55,000	…	…
…	…	…

According to the sentiment lexicons of the bESD, we see that the valences of the sentiment lexicons are from -10 to +10. Thus, a natural part of one valence is presented by the 4 binary bits and we also use the 4 binary bits of a surplus part of this valence. So, the 8 binary bits are used for presenting one valence of one sentiment lexicons in a binary code.

Based on the English dictionaries [33-38], the longest word in English has 189,819 letters. According to the binary code of letters in English in [47-52], we see that the 7 binary bits are used in encode one letter in all the letters in English. Therefore, we need 189,819 (letters) x 7 (bits) = 1,328,733 (bits) to present one word in English.

So, we need (1,328,733 bits of the content + 8 bits of the valence) = 1,328,741 bits to show fully one sentiment lexicon of the bESD in Fig 6 as follows:

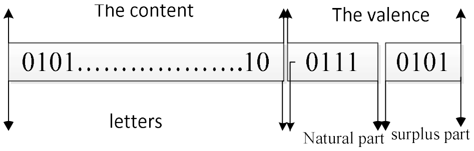

Fig. 6: Overview Of Presenting One Sentiment Lexicon Of The Besd In A Binary Code

4.2.2 Using the sentiment lexicons to transfer all the sentences of one document of the testing data set and all the sentences of the training data set into the binary bits strings in a sequential system

In this section, we use the sentiment lexicons to transfer all the sentences of one document of the testing data set and all the sentences of the training data set into the binary bits strings in a sequential system according to the displaying the theories of transferring one sentence into one binary bits string (4.2.1)

We built the algorithm 4 to encrypt one sentiment lexicon (comprising the content and the valence) to a binary array in the sequential environment.

```
Input: one sentiment lexicon of the bESD
Output: a bit array
Step 1: Split this term into the letters.
Step 2: Set ABitArray := null;
Step 3: Set Valence := Get a valence of this term based on the bESD;
Step 4: Each letter in the letters, do repeat:
Step 5: Based on the binary code of letters in English in [47-52], we get a bit array of this letter;
Step 6: Add the bit array of this letter into ABitArray;
Step 7: End Repeat – End Step 3;
Step 8: Based on on the transferring a decimal to a binary code in [53-58], we transfer the valence to a bit array;
Step 9: Add this bit array into ABitArray;
Step 10: Return ABitArray;
```

We proposed the algorithm 5 to encode one sentence in English to a binary array in the sequential system.

```
Input: one sentence;
Output: a bit array;
Step 1: Set ABitArrayOfSentence := null;
Step 2: Split this sentence into the meaningful terms (meaningful word or meaningful phrase);
Step 3: Each term in the terms, do repeat:
Step 4: ABitArray := The algorithm 4 to encrypt one sentiment lexicon (comprising the content and the valence) to a binary array in the sequential environment with the input is this term;
Step 5: Add ABitArray into ABitArrayOfSentence;
Step 6: End Repeat – End Step 3;
Step 7: Return ABitArrayOfSentence;
```

We built the algorithm 6 to encrypt all the positive sentences of the training data set to the positive bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the sequential environment, called the positive group.

```
Input: all the positive sentences of the training data set
Output:a positive bit array group;
Step 1: Set APositiveBitArrayGroup := null;
Step 2: Each sentence in the positive sentences, do repeat:
Step 3: ABitArray := the algorithm 5 to encode one sentence in English to a binary array in the sequential system with the input is this sentence;
Step 4: Add ABitArray into APositiveBitArrayGroup;
Step 5: End Repeat – End Step 2;
Step 6: Return APositiveBitArrayGroup;
```

We proposed the algrotihm 7 to encode all the negative sentences of the training data set to the negative bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the sequential system, called the negative group.

```
Input: all the negative sentences of the training data set
Output:a negative bit array group;
Step 1: Set ANegativeBitArrayGroup := null;
Step 2: Each sentence in the positive sentences, do repeat:
Step 3: ABitArray := the algorithm 5 to encode one sentence in English to a binary array in the sequential system with the input is this sentence;
Step 4: Add ABitArray into ANegativeBitArrayGroup;
Step 5: End Repeat – End Step 2;
Step 6: Return ANegativeBitArrayGroup;
```

We built the algorithm 8 to transfer one document of the testing data set into the bit arrays of the document in the sequential system.

```
Input: one document of the testing data set
Output: the bit arrays of the document;
Step 1: Set TheBitArraysOfTheDocument := null;
Step 2: Split this document into the sentences;
Step 3: Each sentence in the sentences, do repeat:
Step 4: ABitArray := the algorithm 5 to encode one sentence in English to a binary array in the sequential system with the input is this sentence;
Step 5: Add ABitArray into TheBitArraysOfTheDocument;
Step 6: End Repeat- End Step 3;
Step 7: Return TheBitArraysOfTheDocument;
```

4.2.3 Using the sentiment lexicons to transfer all the sentences of one document of the testing data set and all the sentences of the training data set into the binary bits strings in a parallel network environment

In this section, we use the sentiment lexicons to transfer all the sentences of one document of the testing data set and all the sentences of the training data set into the binary bits strings in a parallel system based on the displaying the theories of transferring one sentence into one binary bits string (4.2.1)

In Fig 7, we build the algorithm 9 and the algorithm 10 to encrypt one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment. This stage in Fig 7 comprises two phases as follows: Hadoop Map phase and Hadoop Reduce phase. The input of the Hadoop Map is one sentiment lexicon of the bESD. The output of the Hadoop Map is a bit array of one letter. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is a bit array of one letter. The output of the Hadoop Reduce is a bit array of the term.

We proposed the algorithm 9 to implement the Hadoop Map phase

Input: one sentiment lexicon of the bESD
Output: a bit array of one letter;
Step 1: Input this term and the bESD into the Hadoop Map in the Cloudera system;
Step 2: Split this term into the letters.
Step 3: Set Valence := Get a valence of this term based on the bESD;
Step 4: Each letter in the letters, do repeat:
Step 5: Based on the binary code of letters in English in [47-52], we get a bit array of this letter;
Step 6: Return the bit array of this letter; //the output of the Hadoop Map

We built the algorithm 10 to perform the Hadoop Reduce phase

Input: the bit array of this letter; //the output of the Hadoop Map
Output: a bit array of the term - ABitArray;
Step 1: Receive the bit array of this letter;
Step 2: Add the bit array of this letter into ABitArray;
Step 3: If this term is full Then
Step 4: Based on on the transferring a decimal to a binary code in [53-58], we transfer the valence to a bit array;
Step 5: Add this bit array into ABitArray;
Step 6: End If – End Step 3;
Step 7: Return ABitArray;

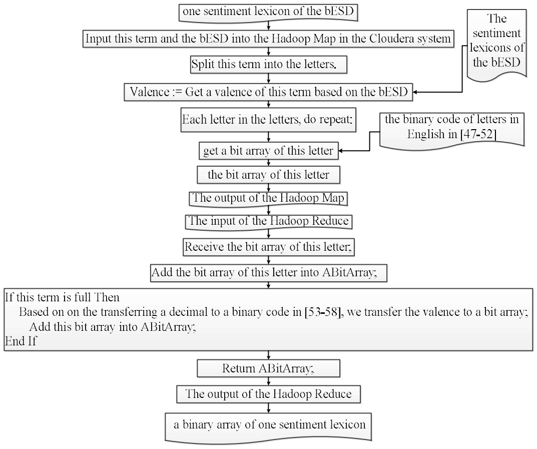
one sentiment lexicon of the bESD
Input this term and the bESD into the Hadoop Map in the Cloudera system
The sentiment lexicons of the bESD
Split this term into the letters.
Valence := Get a valence of this term based on the bESD
Each letter in the letters, do repeat:
the binary code of letters in English in [47-52]
get a bit array of this letter
the bit array of this letter
The output of the Hadoop Map
The input of the Hadoop Reduce
Receive the bit array of this letter;
Add the bit array of this letter into ABitArray;
If this term is full Then
Based on on the transferring a decimal to a binary code in [53-58], we transfer the valence to a bit array;
Add this bit array into ABitArray;
End If
Return ABitArray;
The output of the Hadoop Reduce
a binary array of one sentiment lexicon

Fig. 7: Overview Of Encrypting One Sentiment Lexicon (Comprising The Content And The Valence) To A Binary Array In The Distributed Network Environment

In Fig 8, we build the algorithm 11 and the algorithm 12 to encode one sentence in English to a binary array in the distributed system. This stage in Fig 8 comprises two phases as follows: Hadoop Map phase and Hadoop Reduce phase. The input of the Hadoop Map is one sentence. The output of the Hadoop Map is a bit array of one term - ABitArray. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is a bit array of one term - ABitArray. The output of the Hadoop Reduce is a bit array of the sentence – AbitArrayOfSentence.

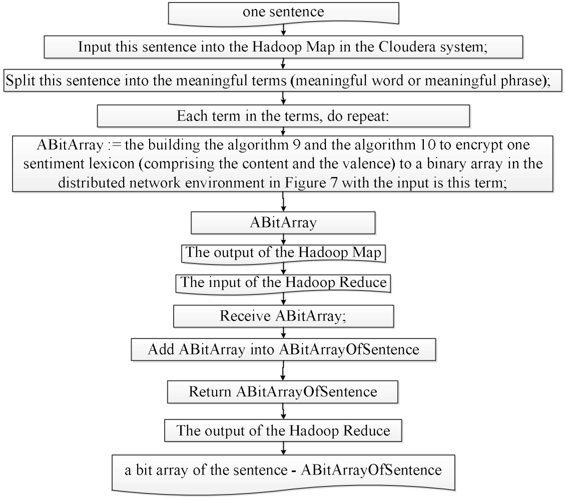
one sentence
Input this sentence into the Hadoop Map in the Cloudera system;
Split this sentence into the meaningful terms (meaningful word or meaningful phrase);
Each term in the terms, do repeat:
ABitArray := the building the algorithm 9 and the algorithm 10 to encrypt one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment in Figure 7 with the input is this term;
ABitArray
The output of the Hadoop Map
The input of the Hadoop Reduce
Receive ABitArray;
Add ABitArray into ABitArrayOfSentence
Return ABitArrayOfSentence
The output of the Hadoop Reduce
a bit array of the sentence - ABitArrayOfSentence

Fig. 8: Overview Of Encoding One Sentence In English To A Binary Array In The Parallel System

We proposed the algorithm 11 to perform the Hadoop Map phase

Input: one sentence;
Output: a bit array of one term - ABitArray;
Step 1: Input this sentence into the Hadoop Map in the Cloudera system;

Step 2: Split this sentence into the meaningful terms (meaningful word or meaningful phrase);
Step 3: Each term in the terms, do repeat:
Step 4: ABitArray := the building the algorithm 9 and the algorithm 10 to encrypt one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment in Fig 7 with the input is this term;
Step 5: Return ABitArray;

We built the algorithm 12 to implement the Hadoop Reduce of encoding one sentence in English to a binary array in the parallel system.
Input: a bit array of one term – AbitArray – the output of the Hadoop Map;
Output: a bit array of the sentence - ABitArrayOfSentence;
Step 1: Receive AbitArray;
Step 2: Add ABitArray into ABitArrayOfSentence;
Step 3: Return ABitArrayOfSentence;

In Fig 9, we build the algorithm 13 and the algorithm 14 to encrypt all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the positive bit array group. This stage comprises two phases as follows: Hadoop Map phase and Hadoop Reduce phase. The input of the Hadoop Map is all the positive sentences of the training data set. The output of the Hadoop Map is ABitArray – a bit array of one sentence. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is ABitArray – a bit array of one sentence. The output of the Hadoop Reduce is a positive bit array group – APositiveGroup.

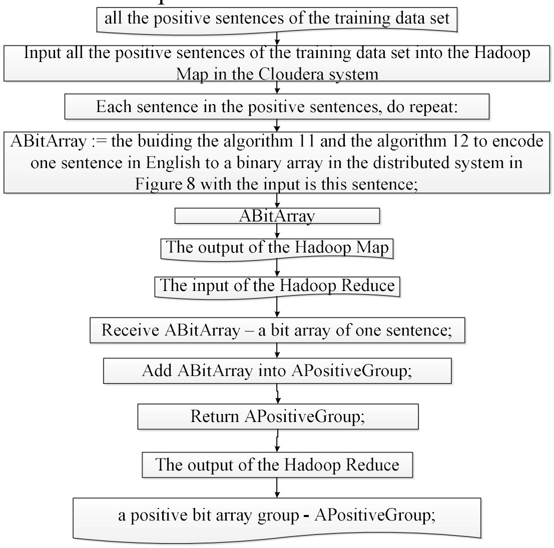
all the positive sentences of the training data set
Input all the positive sentences of the training data set into the Hadoop Map in the Cloudera system
Each sentence in the positive sentences, do repeat:
ABitArray := the buiding the algorithm 11 and the algorithm 12 to encode one sentence in English to a binary array in the distributed system in Figure 8 with the input is this sentence;
ABitArray
The output of the Hadoop Map
The input of the Hadoop Reduce
Receive ABitArray – a bit array of one sentence;
Add ABitArray into APositiveGroup;
Return APositiveGroup;
The output of the Hadoop Reduce
a positive bit array group - APositiveGroup;

Fig. 9: Overview Of Encrypting All The Positive Sentences Of The Training Data Set To The Bit Arrays

Based On The Bit Arrays Of The Sentiment Lexicons Of The Besd In The Distributed Environment, Called The Positive Group

We proposed the algorithm 13 to implement the Hadoop Map phase
Input: all the positive sentences of the training data set
Output: ABitArray – a bit array of one sentence;
Step 1: Input all the positive sentences of the training data set into the Hadoop Map in the Cloudera system;
Step 2: Each sentence in the positive sentences, do repeat:
Step 3: ABitArray := the buiding the algorithm 11 and the algorithm 12 to encode one sentence in English to a binary array in the distributed system in Fig 8 with the input is this sentence;
Step 4: Return ABitArray;

We built the algorithm 14 to propose the Hadoop Reduce phase
Input: ABitArray – a bit array of one sentence;
Output:a positive bit array group - APositiveGroup;
Step 1: Receive ABitArray;
Step 2: Add ABitArray into APositiveGroup;
Step 3: Return APositiveGroup;

In Fig 10, we build the algorithm 15 and the algorithm 16 to encrypt all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the negative group. This stage comprises two phases as follows: Hadoop Map phase and Hadoop Reduce phase. The input of the Hadoop Map is all the negative sentences of the training data set. The output of the Hadoop Map is ABitArray – a bit array of one sentence. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is ABitArray – a bit array of one sentence. The output of the Hadoop Reduce is a negative bit array group – ANegativeGroup.

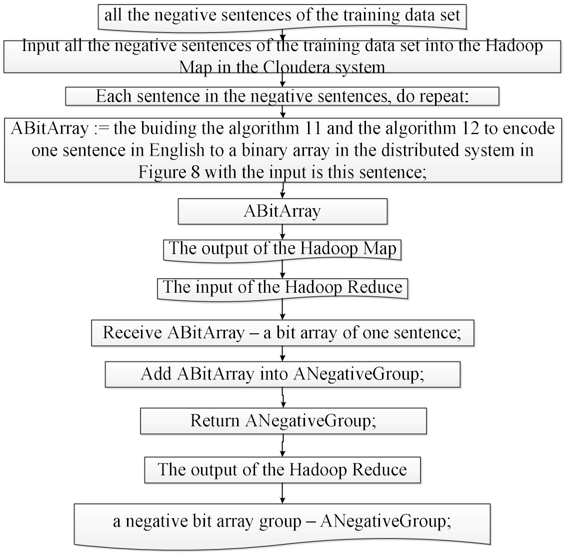
all the negative sentences of the training data set
Input all the negative sentences of the training data set into the Hadoop Map in the Cloudera system
Each sentence in the negative sentences, do repeat:
ABitArray := the buiding the algorithm 11 and the algorithm 12 to encode one sentence in English to a binary array in the distributed system in Figure 8 with the input is this sentence;
ABitArray
The output of the Hadoop Map
The input of the Hadoop Reduce
Receive ABitArray – a bit array of one sentence;
Add ABitArray into ANegativeGroup;
Return ANegativeGroup;
The output of the Hadoop Reduce
a negative bit array group – ANegativeGroup;

Fig. 10: Overview Of Encrypting All The Negative Sentences Of The Training Data Set To The Bit Arrays Based On The Bit Arrays Of The Sentiment Lexicons Of The Besd In The Distributed Environment, Called The Negative Group

We proposed the algorithm 15 to implement the Hadoop Map phase

Input: all the negative sentences of the training data set

Output: ABitArray – a bit array of one sentence;

Step 1: Input all the negative sentences of the training data set into the Hadoop Map in the Cloudera system;

Step 2: Each sentence in the negative sentences, do repeat:

Step 3: ABitArray := the buiding the algorithm 11 and the algorithm 12 to encode one sentence in English to a binary array in the distributed system in Fig 8 with the input is this sentence;

Step 4: Return ABitArray;

We built the algorithm 16 to propose the Hadoop Map phase

Input: ABitArray – a bit array of one sentence;

Output:a negative bit array group - ANegativeGroup;

Step 1: Receive ABitArray;

Step 2: Add ABitArray into ANegativeGroup;

Step 3: Return ANegativeGroup;

In Fig 11, we build the algorithm 17 and the algorithm 18 to transfer one document of the testing data set into the bit arrays of the document in the parallel system. This stage comprises two phases as follows: Hadoop Map phase and Hadoop Reduce phase. The input of the Hadoop Map is one document of the testing data set. The output of the Hadoop Map is one bit array of one sentence of the document – the output of the Hadoop Map. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is one bit array of one sentence of the document – the output of the Hadoop Map. The output of the Hadoop Reduce is the bit arrays of the document.

We proposed the algorithm 17 to perform the Hadoop Map phase

Input: one document of the testing data set

Output: ABitArray - one bit array of one sentence of the document – the output of the Hadoop Map;

Step 1: Input one document of the testing data set 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: ABitArray := the buiding the algorithm 11 and the algorithm 12 to encode one sentence in English to a binary array in the distributed system in Fig 8 with the input is this sentence;

Step 5: Return ABitArray;

We built the algorithm 18 to perform the Hadoop Reduce phase

Input: ABitArray - one bit array of one sentence of the document – the output of the Hadoop Map;

Output: the bit arrays of the document - TheBitArraysOfTheDocument;

Step 1:Receive ABitArray;

Step 2: Add ABitArray into TheBitArraysOfTheDocument;

Step 3: Return TheBitArraysOfTheDocument;

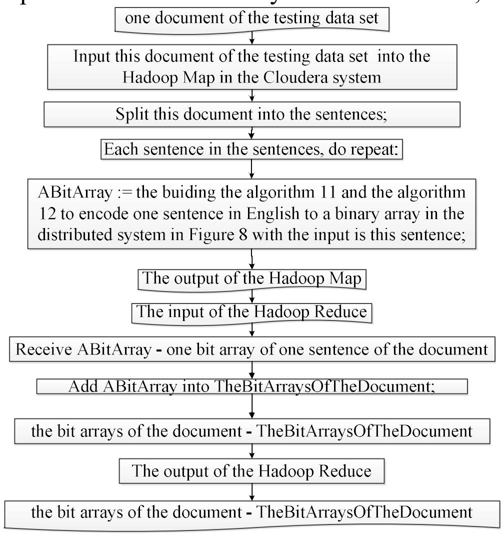
one document of the testing data set
Input this document of the testing data set into the Hadoop Map in the Cloudera system
Split this document into the sentences;
Each sentence in the sentences, do repeat:
ABitArray := the buiding the algorithm 11 and the algorithm 12 to encode one sentence in English to a binary array in the distributed system in Figure 8 with the input is this sentence;
The output of the Hadoop Map
The input of the Hadoop Reduce
Receive ABitArray - one bit array of one sentence of the document
Add ABitArray into TheBitArraysOfTheDocument;
the bit arrays of the document - TheBitArraysOfTheDocument
The output of the Hadoop Reduce
the bit arrays of the document - TheBitArraysOfTheDocument

Fig. 11: Overview Of Transferring One Document Of The Testing Data Set Into The Bit Arrays Of The Document In The Parallel System

4.3 Using the binary bits and sentiment lexicons to classify the sentiments (positive, negative, or neutral) for the documents of the testing data set in both the sequential environment and the distributed system.

This section has two sub-sections as follows: (4.3.1) and (4.3.2). In the sub-section (4.3.1), we use the binary bits and sentiment lexicons to classify the sentiments (positive, negative, or neutral) for the documents of the testing data set in a sequential environment. In the sub-section (4.3.2), we use the binary bits and sentiment lexicons to classify the sentiments (positive, negative, or neutral) for the documents of the testing data set in a distributed system.

4.3.1 Calculating the valence of the sentiment lexicons

In this section, we use the binary bits and sentiment lexicons to classify the documents of the testing data set into either the positive polarity or the negative polarity in a sequential environment as follows: Based on the sentiment lexicons of the bESD in the sequential environment (4.1.2), we firlsty identify the sentiment scores and polarities of the sentiment lexicons of the bESD by using the YSC through the Google search engine with AND operator and OR operator. According to the surveys related to the binary code of letters in English in [47-52] and the researches related to transferring a decimal to a binary code in [53-58], we transfer one sentence into one binary bits string. One binary bits string is one binary bits vector with each element of this vector is a bit 0 or 1. A positive group is a group which we transfer all the positive sentences of the training data set into the positive binary bits strings and a negative group is a group which we also transfer all the negative sentences of the training data set into the negative binary bits strings. All the sentences of one document of the testing data set are transferred into the binary bits strings. Each binary bits string in the binary bits strings of this document, we calculate a total of all the similarity measures of this string with all the positive binary bits string of the positive group by using the YSC, called ATotalOfPositiveSimilarityMeasures, and we also calculate a total of all the similarity measures of this string with all the negative binary bits string of the negative group by using the YSC, called ATotalOfNagativeSimilarityMeasures. This binary bits string is clustered into the positive if ATotalOfPositiveSimilarityMeasures is greater than ATotalOfNegativeSimilarityMeasures. This binary bits string is clustered into the negative if ATotalOfPositiveSimilarityMeasures is less than ATotalOfNegativeSimilarityMeasures. This binary bits string is clustered into the neutral if ATotalOfPositiveSimilarityMeasures is as equal as ATotalOfNegativeSimilarityMeasures. One document is clustered into the positive if the number of the binary bits strings clustered into the positive is greater than that clustered into the negative in this document. The document is clustered into the negative if the number of the binary bits strings clustered into the positive is less than that clustered into the negative in this document. One document is clustered into the neutral if the number of the binary bits strings clustered into the positive is as equal as that clustered into the negative in this document. Finally, the sentiment classification of all the documents of the testing data set is identified certainly.

We built the algorithm 19 to identify a total of all the similarity measures of one binary bits string of one document of the testing data set with all the positive binary bits string of the positive group by using the YSC, called ATotalOfPositiveSimilarityMeasures.

```
Input: one binary bits string A of one document of the testing data set and a positive group of the training data set;
Output: a total of all the similarity measures - ATotalOfPositiveSimilarityMeasures;
Step 1: Set ATotalOfPositiveSimilarityMeasures := 0;
Step 2: This binary bits string A is a binary bits vector with each element is a bit 0 or 1;
Step 3: Each binary bits vector B in the positive group, repeat:
Step 4: SimilarityMeasure := Calculate a similarity measure of this vector A with vector B by using eq. (3) of the calculating the valence of one sentiment lexicon (4.1.1);
Step 5: ATotalOfPositiveSimilarityMeasures := ATotalOfPositiveSimilarityMeasures + SimilarityMeasure;
Step 6: End Repeat – End Step 3;
Step 7: Return ATotalOfPositiveSimilarityMeasures;
```

We proposed the algorithm 20 to identify a total of all the similarity measures of one binary bits string of one document of the testing data set with all the negative binary bits string of the negative group by using the YSC, called ATotalOfNegativeSimilarityMeasures.

```
Input: one binary bits string A of one document of the testing data set and a negative group of the training data set;
Output: a total of all the similarity measures - ATotalOfNegativeSimilarityMeasures;
Step 1: Set ATotalOfNegativeSimilarityMeasures := 0;
Step 2: This binary bits string A is a binary bits vector with each element is a bit 0 or 1;
Step 3: Each binary bits vector B in the negative group, repeat:
    Step 4: SimilarityMeasure := Calculate a similarity measure of this vector A with vector B by using eq. (3) of the calculating the valence of one sentiment lexicon (4.1.1);
    Step 5: ATotalOfNegativeSimilarityMeasures := ATotalOfNegativeSimilarityMeasures + SimilarityMeasure;
Step 6: End Repeat – End Step 3;
Step 7: Return ATotalOfNegativeSimilarityMeasures;
```

We built the algorithm 21 to identify the sentiment classification of one binary bits string of the testing data set in the sequential environment.

```
Input: one binary bits string of the testing data set; the positive group and the negative group of the training data set;
Output: positive, negative, or neutral;
Step 1: Set ATotalOfPositiveSimilarityMeasures := the algorithm 19 to identify a total of all the similarity measures of one binary bits string of one document of the testing data set with all the positive binary bits string of the positive group by using the YSC, called ATotalOfPositiveSimilarityMeasures.
Step 2: Set ATotalOfNegativeSimilarityMeasures := the algorithm 20 to identify a total of all the similarity measures of one binary bits string of one document of the testing data set with all the negative binary bits string of the negative group by using the YSC, called ATotalOfNegativeSimilarityMeasures
Step 3: If ATotalOfPositiveSimilarityMeasures is greater than ATotalOfNegativeSimilarityMeasuresThen Return Positive;
Step 4: If ATotalOfPositiveSimilarityMeasures is less than ATotalOfNegativeSimilarityMeasures Then Return Negative;
Step 5: Return Neutral;
```

We proposed the algorithm 22 to identify the sentiment classification of one document of the testing data set in the sequential environment

```
Input: one document of the testing data set; the positive group and the negative group of the training data set;
Output: positive, negative, or neutral;
Step 1: Split this document into the sentences;
Step 2: Set PositiveNumber := 0 and NegativeNumber := 0;
Step 3:Each sentence in the sentences, repeat:
    Step 4: OneBinaryBitsVector := the algorithm 5 to encode one sentence in English to a binary array in the sequential system with the input is this sentence;
    Step 5: OneResult := the algorithm 21 to identify the sentiment classification of one binary bits string of the testing data set in the sequential environment with the input is OneBinaryBitsVector; the positive group and the negative group;
    Step 6: If OneResult is positive Then PositiveNumber := PositiveNumber +1 ;
    Step 7: Else If OneResult is negative Then NegativeNumber := NegativeNumber +1 ;
Step 8: If PositiveNumber is greater than NegativeNumber Then Return Positive;
Step 9: Else If PositiveNumber is less than NegativeNumber Then Return Negative;
Step 10: Return Neutral;
```

We built the algorithm 23 to identify the sentiment classification of all the documents of the testing data set in the sequential system.

```
Input: the documents of the testing data set and the sentences of the training data set;
Output: positive, negative, or neutral;
Step 1: the sentiment lexicons of the bESD in the sequential environment (4.1.2)
Step 2: the algorithm 6 to encrypt all the positive sentences of the training data set to the positive bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the sequential environment, called the positive group.
Step 3: the algrotihm 7 to encode all the negative sentences of the training data set to the negative bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the sequential system, called the negative group
Step 4: Set TheResults := null;
Step 5: Each document in the documents of the testing data set, repeat:
    Step 6: OneResult := the algorithm 22 to identify the sentiment classification of one document of the testing data set in the sequential environment.
    Step 7: Add OneResult into TheResults;
Step 8: End Repeat – End Step 5;
Step 9: Return TheResults;
```

4.3.2 Using the binary bits and sentiment lexicons to classify the sentiments (positive, negative, or neutral) for the documents of the testing data set in the distributed system.

In this section, we use the binary bits and sentiment lexicons to classify the documents of the testing data set into either the positive polarity or the negative polarity in a parallel network environment as follows: Based on the creating a basis English sentiment dictionary (bESD) in a distributed environment (4.1.3), we firlsty identify the sentiment scores and polarities of the sentiment lexicons of the bESD by using the YSC through the Google search engine with AND operator and OR operator. According to the surveys related to the binary code of letters in English in [47-52] and the researches related to transferring a decimal to a binary code in [53-58], we transfer one sentence into one binary bits string. One binary bits string is one binary bits vector with each element of this vector is a bit 0 or 1. A positive group is a group which we transfer all the positive sentences of the training data set into the positive binary bits strings and a negative group is a group which we also transfer all the negative sentences of the training data set into the negative binary bits strings. All the sentences of one document of the testing data set are transferred into the binary bits strings. Each binary bits string in the binary bits strings of this document, we calculate a total of all the similarity measures of this string with all the positive binary bits string of the positive group by using the YSC, called ATotalOfPositiveSimilarityMeasures, and we also calculate a total of all the similarity measures of this string with all the negative binary bits string of the negative group by using the YSC, called ATotalOfNagativeSimilarityMeasures. This binary bits string is clustered into the positive if ATotalOfPositiveSimilarityMeasures is greater than ATotalOfNegativeSimilarityMeasures. This binary bits string is clustered into the negative if ATotalOfPositiveSimilarityMeasures is less than ATotalOfNegativeSimilarityMeasures. This binary bits string is clustered into the neutral if ATotalOfPositiveSimilarityMeasures is as equal as ATotalOfNegativeSimilarityMeasures. One document is clustered into the positive if the number of the binary bits strings clustered into the positive is greater than that clustered into the negative in this document. The document is clustered into the negative if the number of the binary bits strings clustered into the positive is less than that clustered into the negative in this document. One document is clustered into the neutral if the number of the binary bits strings clustered into the positive is as equal as that clustered into the negative in this document. Finally, the sentiment classification of all the documents of the testing data set is identified certainly.

In Fig 12, we calculate a total of all the similarity measures of one binary bits string with all the positive binary bits string of the positive group by using the YSC, called ATotalOfPositiveSimilarityMeasures. This stage includes two phases as follows: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map phase is one binary bits string. The output of the Hadoop Map phase is one similarity measure. The input of the Hadoop Reduce phase is the output of the Hadoop Map, thus, the input of the Hadoop Reduce phase is one similarity measure. The output of the Hadoop Reduce phase is a total of all the similarity measures - ATotalOfPositiveSimilarityMeasures

We built the algorithm 24 to perform the Hadoop Map phase

Input: one binary bits string A of one document of the testing data set and a positive group of the training data set;

Output: one similarity measure;

Step 1: Input this binary bits string A of one document of the testing data set and a positive group of the training data set into the Hadoop Map in the Cloudera system;

Step 2: This binary bits string A is a binary bits vector with each element is a bit 0 or 1;

Step 3: Each binary bits vector B in the positive group, repeat:

Step 4: SimilarityMeasure := Calculate a similarity measure of this vector A with vector B by using eq. (3) of the calculating the valence of one sentiment lexicon (4.1.1);

Step 5: Return SimilarityMeasure;

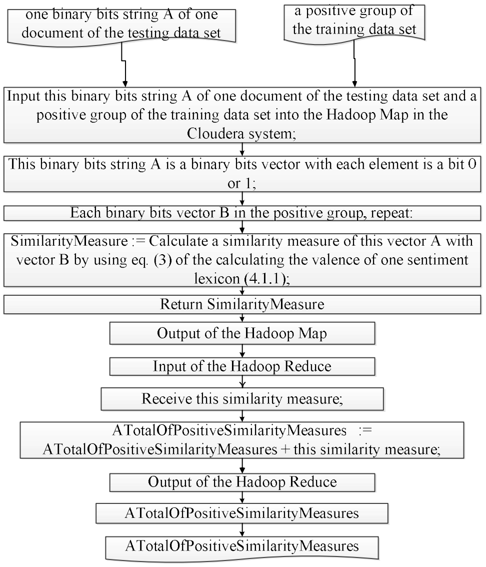
one binary bits string A of one document of the testing data set
a positive group of the training data set
Input this binary bits string A of one document of the testing data set and a positive group of the training data set into the Hadoop Map in the Cloudera system;
This binary bits string A is a binary bits vector with each element is a bit 0 or 1;
Each binary bits vector B in the positive group, repeat:
SimilarityMeasure := Calculate a similarity measure of this vector A with vector B by using eq. (3) of the calculating the valence of one sentiment lexicon (4.1.1);
Return SimilarityMeasure
Output of the Hadoop Map
Input of the Hadoop Reduce
Receive this similarity measure;
ATotalOfPositiveSimilarityMeasures := ATotalOfPositiveSimilarityMeasures + this similarity measure;
Output of the Hadoop Reduce
ATotalOfPositiveSimilarityMeasures
ATotalOfPositiveSimilarityMeasures

Fig. 12: Overview Of Calculating A Total Of All The Similarity Measures Of One Binary Bits String With All The Positive Binary Bits String Of The Positive Group By Using The YSC, Called Atotalofpositivesimilaritymeasures

We proposed the algorithm 25 to perform the Hadoop Reduce phase

Input: one similarity measure; //the output of the Hadoop Map

Output: a total of all the similarity measures - ATotalOfPositiveSimilarityMeasures

Step 1: Receive this similarity measure;

Step 2: ATotalOfPositiveSimilarityMeasures := ATotalOfPositiveSimilarityMeasures + this similarity measure;

Step 3: Return ATotalOfPositiveSimilarityMeasures;

In Fig 13, we calculate a total of all the similarity measures of one binary bits string with all the positive binary bits string of the negative group by using the YSC, called ATotalOfNegativeSimilarityMeasures. This stage includes two phases as follows: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map phase is one binary bits string. The output of the Hadoop Map phase is one similarity measure. The input of the Hadoop Reduce phase is the output of the Hadoop Map, thus, the input of the Hadoop Reduce phase is one similarity measure. The output of the Hadoop Reduce phase is a total of all the similarity measures - ATotalOfNegativeSimilarityMeasures

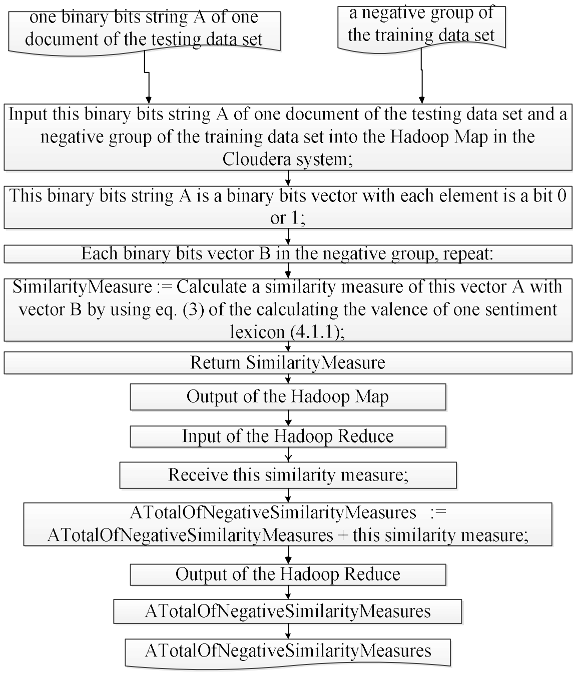
one binary bits string A of one document of the testing data set
a negative group of the training data set
Input this binary bits string A of one document of the testing data set and a negative group of the training data set into the Hadoop Map in the Cloudera system;
This binary bits string A is a binary bits vector with each element is a bit 0 or 1;
Each binary bits vector B in the negative group, repeat:
SimilarityMeasure := Calculate a similarity measure of this vector A with vector B by using eq. (3) of the calculating the valence of one sentiment lexicon (4.1.1);
Return SimilarityMeasure
Output of the Hadoop Map
Input of the Hadoop Reduce
Receive this similarity measure;
ATotalOfNegativeSimilarityMeasures := ATotalOfNegativeSimilarityMeasures + this similarity measure;
Output of the Hadoop Reduce
ATotalOfNegativeSimilarityMeasures
ATotalOfNegativeSimilarityMeasures

Fig. 13: Overview Of Calculating A Total Of All The Similarity Measures Of One Binary Bits String With All The Negative Binary Bits String Of The Negative Group By Using The YSC, Called Atotalofnegativesimilaritymeasures

We built the algorithm 26 to perform the Hadoop Map phase

Input: one binary bits string A of one document of the testing data set and a negative group of the training data set;

Output: one similarity measure;

Step 1: Input this binary bits string A of one document of the testing data set and a negative group of the training data set into the Hadoop Map in the Cloudera system;

Step 2: This binary bits string A is a binary bits vector with each element is a bit 0 or 1;

Step 3: Each binary bits vector B in the negative group, repeat:

Step 4: SimilarityMeasure := Calculate a similarity measure of this vector A with vector B by using eq. (3) of the calculating the valence of one sentiment lexicon (4.1.1);

Step 5: Return SimilarityMeasure;

We proposed the algorithm 27 to perform the Hadoop Reduce phase

Input: one similarity measure; //the output of the Hadoop Map

Output: a total of all the similarity measures - ATotalOfNegativeSimilarityMeasures

Step 1: Receive this similarity measure;

Step 2: ATotalOfNegativeSimilarityMeasures := ATotalOfNegativeSimilarityMeasures + this similarity measure;
Step 3: Return ATotalOfNegativeSimilarityMeasures;

In Fig 14, we identify the sentiment classification of one binary bits string of the testing data set in the distributed environment. This stage includes two phases as follows: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map phase is one document of the testing data set. The output of the Hadoop Map phase is OneResult – the result of the sentiment classification of this document. The input of the Hadoop Reduce phase is the output of the Hadoop Map, thus, the input of the Hadoop Reduce phase is OneResult – the result of the sentiment classification of this document. The output of the Hadoop Reduce phase is the result of the sentiment classification of this document.

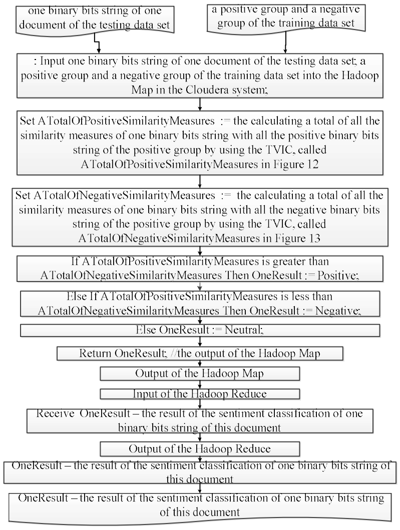
one binary bits string of one document of the testing data set
a positive group and a negative group of the training data set
: Input one binary bits string of one document of the testing data set; a positive group and a negative group of the training data set into the Hadoop Map in the Cloudera system;
Set ATotalOfPositiveSimilarityMeasures := the calculating a total of all the similarity measures of one binary bits string with all the positive binary bits string of the positive group by using the TVIC, called ATotalOfPositiveSimilarityMeasures in Figure 12
Set ATotalOfNegativeSimilarityMeasures := the calculating a total of all the similarity measures of one binary bits string with all the negative binary bits string of the positive group by using the TVIC, called ATotalOfNegativeSimilarityMeasures in Figure 13
If ATotalOfPositiveSimilarityMeasures is greater than ATotalOfNegativeSimilarityMeasures Then OneResult := Positive;
Else If ATotalOfPositiveSimilarityMeasures is less than ATotalOfNegativeSimilarityMeasures Then OneResult := Negative;
Else OneResult := Neutral;
Return OneResult; //the output of the Hadoop Map
Output of the Hadoop Map
Input of the Hadoop Reduce
Receive OneResult – the result of the sentiment classification of one binary bits string of this document
Output of the Hadoop Reduce
OneResult – the result of the sentiment classification of one binary bits string of this document
OneResult – the result of the sentiment classification of one binary bits string of this document

Fig. 14: Overview Of Identifying The Sentiment Classification Of One Binary Bits String Of One Document Of The Testing Data Set In The Distributed Environment.

We built the algorithm 28 to perform the Hadoop Map phase

Input: one binary bits string of one document of the testing data set; a positive group and a negative group of the training data set;
Output: OneResult – the result of the sentiment classification of one binary bits string of this document;//the output of the Hadoop Map
Step 1: Input one binary bits string of one document of the testing data set; a positive group and a negative group of the training data set into the Hadoop Map in the Cloudera system;
Step 2: Set ATotalOfPositiveSimilarityMeasures := the calculating a total of all the similarity measures of one binary bits string with all the positive binary bits string of the positive group by using the YSC, called ATotalOfPositiveSimilarityMeasures in Fig 12
Step 3: Set ATotalOfNegativeSimilarityMeasures := the calculating a total of all the similarity measures of one binary bits string with all the negative binary bits string of the positive group by using the YSC, called ATotalOfNegativeSimilarityMeasures in Fig 13
Step 3: If ATotalOfPositiveSimilarityMeasures is greater than ATotalOfNegativeSimilarityMeasures Then OneResult := Positive;
Step 4: Else If ATotalOfPositiveSimilarityMeasures is less than ATotalOfNegativeSimilarityMeasures Then OneResult := Negative;
Step 5: Else OneResult := Neutral;
Step 6: Return OneResult; //the output of the Hadoop Map

We proposed the algorithm 29 to perform the Hadoop Reduce phase

Input: OneResult – the result of the sentiment classification of one binary bits string of this document;//the output of the Hadoop Map
Output: positive, negative, or neutral;
Step 1: Receive OneResult;
Step 2: Return OneResult;

In Fig 15, we identify the sentiment classification of one document of the testing data set in the distributed environment. This stage includes two phases as follows: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map phase is one document of the testing data set. The output of the Hadoop Map phase is OneResult – the result of the sentiment classification of one binary bits string of this document. The input of the Hadoop Reduce phase is the output of the Hadoop Map, thus, the input of the Hadoop Reduce phase is OneResult – the result of the sentiment classification of one binary bits string of this document. The output of the Hadoop Reduce phase is the result of the sentiment classification of this document.

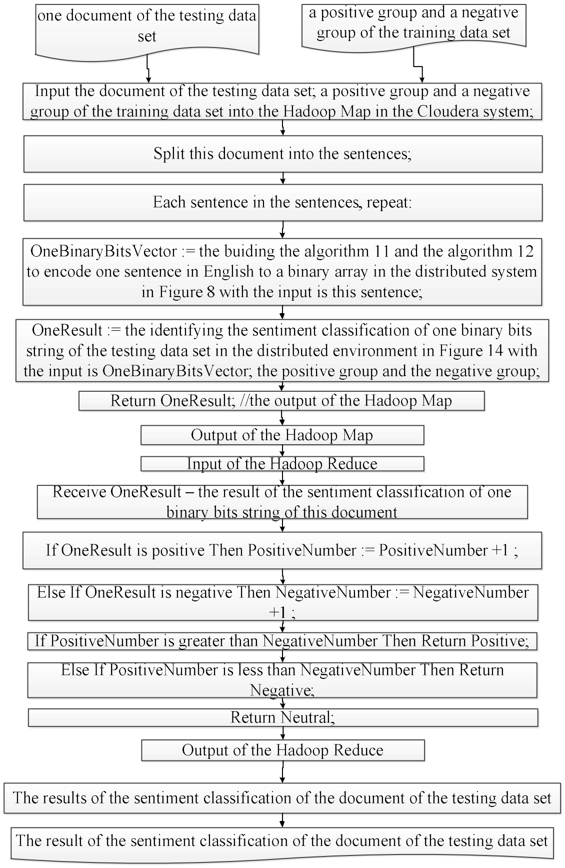
one document of the testing data set
a positive group and a negative group of the training data set
Input the document of the testing data set; a positive group and a negative group of the training data set into the Hadoop Map in the Cloudera system;
Split this document into the sentences;
Each sentence in the sentences, repeat:
OneBinaryBitsVector := the buiding the algorithm 11 and the algorithm 12 to encode one sentence in English to a binary array in the distributed system in Figure 8 with the input is this sentence;
OneResult := the identifying the sentiment classification of one binary bits string of the testing data set in the distributed environment in Figure 14 with the input is OneBinaryBitsVector; the positive group and the negative group;
Return OneResult; //the output of the Hadoop Map
Output of the Hadoop Map
Input of the Hadoop Reduce
Receive OneResult – the result of the sentiment classification of one binary bits string of this document
If OneResult is positive Then PositiveNumber := PositiveNumber +1 ;
Else If OneResult is negative Then NegativeNumber := NegativeNumber +1 ;
If PositiveNumber is greater than NegativeNumber Then Return Positive;
Else If PositiveNumber is less than NegativeNumber Then Return Negative;
Return Neutral;
Output of the Hadoop Reduce
The results of the sentiment classification of the document of the testing data set
The result of the sentiment classification of the document of the testing data set

Fig. 15: Overview Of Identifying The Sentiment Classification Of One Document Of The Testing Data Set In The Distributed Environment.

We built the algorithm 30 to perform the Hadoop Map phase
Input: one document of the testing data set; a positive group and a negative group of the training data set;
Output: OneResult – the result of the sentiment classification of one binary bits string of this document;//the output of the Hadoop Map
Output: positive, negative, or neutral;
Step 1: Input the document of the testing data set; a positive group and a negative group of the training data set into the Hadoop Map in the Cloudera system;
Step 2: Split this document into the sentences;
Step 3: Each sentence in the sentences, repeat:
Step 4: OneBinaryBitsVector := the buiding the algorithm 11 and the algorithm 12 to encode one sentence in English to a binary array in the distributed system in Fig 8 with the input is this sentence;
Step 5: OneResult := the identifying the sentiment classification of one binary bits string of the testing data set in the distributed environment in Fig 14 with the input is OneBinaryBitsVector; the positive group and the negative group;
Step 6: Return OneResult;

We proposed the algorithm 31 to perform the Hadoop Reduce phase
Input: OneResult – the result of the sentiment classification of one binary bits string of this document;//the output of the Hadoop Map
Output: positive, negative, or neutral;
Step 1: Receive OneResult;
Step 2: If OneResult is positive Then PositiveNumber := PositiveNumber +1 ;
Step 3: Else If OneResult is negative Then NegativeNumber := NegativeNumber +1 ;
Step 4: If PositiveNumber is greater than NegativeNumber Then Return Positive;
Step 5: Else If PositiveNumber is less than NegativeNumber Then Return Negative;
Step 6: Return Neutral;

In Fig 16, we identify the sentiment classification of all the documents of the testing data set in the distributed system. This stage includes two phases as follows: 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 Map phase is OneResult – the result of the sentiment classification of one document. The input of the Hadoop Reduce phase is the output of the Hadoop Map, thus, the input of the Hadoop Reduce phase is OneResult – the result of the sentiment classification of one document. The output of the Hadoop Reduce phase is the result of the sentiment classification of the documents of the testing data set

We proposed the algorithm 32 to perform the Hadoop Map phase
Input: the documents of the testing data set and training data set
Output: OneResult – the result of the sentiment classification of one document;//the output of the Hadoop Map
Step 1: the sentiment lexicons of the bESD in the distributed system (4.1.3)
Step 2: the building the algorithm 13 and the algorithm 14 to encrypt all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the positive bit array group in Fig 9
Step 3: the building the algorithm 15 and the algorithm 16 to encrypt all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in

the distributed environment, called the negative group in Fig 10
Step 3: Input the documents of the testing data set; the positive group and the negative group into the Hadoop Map in the Cloudera system;
Step 4: Each document in the documents of the testing data set, repeat:
Step 5: OneResult := the identifying the sentiment classification of one document of the testing data set in the distributed environment in Fig 15
Step 6: Return OneResult;

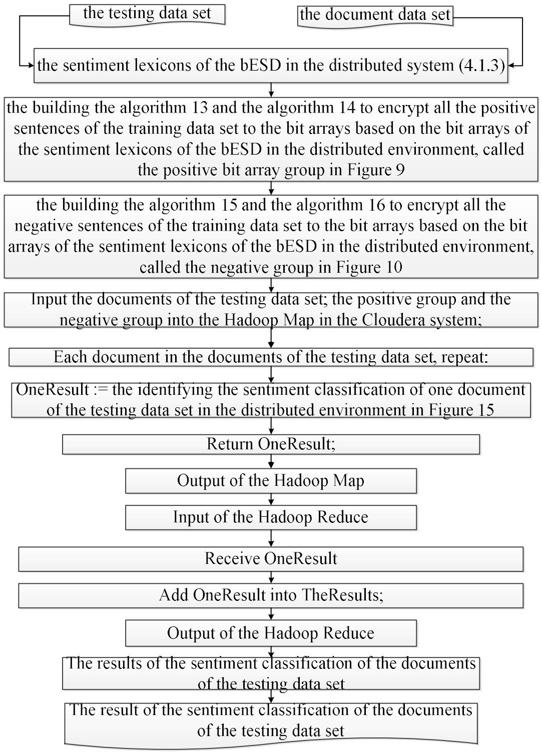
the testing data set
the document data set
the sentiment lexicons of the bESD in the distributed system (4.1.3)
the building the algorithm 13 and the algorithm 14 to encrypt all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the positive bit array group in Figure 9
the building the algorithm 15 and the algorithm 16 to encrypt all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the negative group in Figure 10
Input the documents of the testing data set; the positive group and the negative group into the Hadoop Map in the Cloudera system;
Each document in the documents of the testing data set, repeat:
OneResult := the identifying the sentiment classification of one document of the testing data set in the distributed environment in Figure 15
Return OneResult;
Output of the Hadoop Map
Input of the Hadoop Reduce
Receive OneResult
Add OneResult into TheResults;
Output of the Hadoop Reduce
The results of the sentiment classification of the documents of the testing data set
The result of the sentiment classification of the documents of the testing data set

Fig. 16: Overview Of Identifying The Sentiment Classification Of All The Documents Of The Testing Data Set In The Distributed System.

We proposed the algorithm 33 to perform the Hadoop Reduce phase
Input: OneResult – the result of the sentiment classification of one document;//the output of the Hadoop Map
Output: the result of the sentiment classification of the documents of the testing data set
Step 1: Receive OneResult;
Step 2: Add OneResult into TheResults;
Step 3: Return TheResults;

5. EXPERIMENT

We have measured an Accuracy (A) to calculate the accuracy of the results of emotion classification. A Java programming language is used for programming to save data sets, implementing our proposed model to classify the 9,000,000 documents of the testing 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 sequential environment in this research includes 1 node (1 server). The Java language is used in programming the novel model related to the binary bits and sentiment lexicons. The configuration of the server in the sequential environment is: Intel® Server Board S1200V3RPS, Intel® Pentium® Processor G3220 (3M Cache, 3.00 GHz), 2GB PC3-10600 ECC 1333 MHz LP Unbuffered DIMMs. The operating system of the server is: Cloudera. We perform the novel model related to the binary bits and sentiment lexicons in the Cloudera parallel network environment; this Cloudera system includes 9 nodes (9 servers). The Java language is used in programming the application of the binary bits and sentiment lexicons in the Cloudera. The configuration of each server in the Cloudera system is: Intel® Server Board S1200V3RPS, Intel® Pentium® Processor G3220 (3M Cache, 3.00 GHz), 2GB PC3-10600 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.

In Table 3, we show the results of the English documents in the testing data set.

The accuracy of our new model for the English documents in the testing data set is presented in Table 4.

In Table 5, we display many average execution times of the classification of our new model for the English documents in testing data set.

6. CONCLUSION

Although our new model has been tested on our testing data set, it can be applied to many other languages. In this paper, our model has been tested on the 9,000,000 documents of the testing data set and the 5,000,000 sentences of the training data set which the data sets are small. However, our model can be applied to larger data sets with millions of English documents in the shortest time. In this work, we have proposed a novel model to classify sentiment of English documents using the binary bits and sentiment lexicons with Hadoop Map (M) /Reduce (R) in the Cloudera parallel network environment. With our proposed new model, we have achieved 89.01% accuracy of the testing data set in Table 4. 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 emotion in text.

In Table 5, the average time of the semantic classification of the binary bits and sentiment lexicons in the sequential environment is 35,091,827 seconds /9,000,000 documents and it is greater than the average time of the emotion classification of the binary bits and sentiment lexicons in the Cloudera parallel network environment with 3 nodes which is 10,363,942 seconds / 9,000,000 documents. The average time of the emotion classification of the binary bits and sentiment lexicons in the Cloudera parallel network environment with 9 nodes, which is 3,921,314 seconds /9,000,000 documents, is the shortest time. Besides, The average time of the emotion classification of the binary bits and sentiment lexicons in the Cloudera parallel network environment with 6 nodes is 5,981,971 seconds /9,000,000 documents

The execution time of the binary bits sentiment lexicons 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 proposed model has many advantages and disadvantages. Its positives are as follows: It uses the binary bits and 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. 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 as follows: Table 8, and Table 7.

In Table 8, we show the comparisons of our model with the latest sentiment classification models (or the latest sentiment classification methods) in [59-69]

The comparisons of our model’s positives and negatives the latest sentiment classification models (or the latest sentiment classification methods) in [59-69] are presented in Table 9.

7. 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] 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

[2] 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

[3] Robert Malouf, Tony Mullen, “*Graph-based user classification for informal online political discourse*”, In proceedings of the 1st Workshop on Information Credibility on the Web, 2017

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

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

[6] 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, pp 481–485, 2016

[7] 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, pp 235–243, 2009

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

[9] 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

[10] Tan, S.; Zhang, J. , "*An empirical study of sentiment analysis for Chinese documents*", Expert Systems with Applications (2007), doi:10.1016/j.eswa.2007.05.028, 2007

[1] 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

[12] 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

[13] 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, pp 189–192, 2007

[14] 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, pp 897–902, 2013

[15] 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

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

[17] 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

[18] 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), Volume5, 2013

[19] 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

[20] 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

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

[22] 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, pp 420 – 429, 2011

[23] 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, pp 571-583, 2011

[24] Julia V. Ponomarenko, Philip E. Bourne, Ilya N. Shindyalov, "*Building An Automated Classification Of Dna-Binding Protein Domains"*, Bioinformatics, Vol. 18, Pp S192-S201, 2002

[25] 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, 83-91, 2004

[26] 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, 367- 370, 2008

[27] 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

[28] 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, 67 pages, 2017

[29] 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, 30 pages, 2017

[30] 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, 28 pages, 2017

[31] 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, 47 pages, 2017

[32] 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 Evolving Systems, DOI: 10.1007/s12530-017-9188-6, 69 pages, 2017

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

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

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

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

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

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

[39] Hadoop, *http://hadoop.apache.org*, 2017

[40] Apache, *http://apache.org*, 2017

[41] Cloudera, *http://www.cloudera.com*, 2017

[42] Rodham 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. Mycology in Sustainable Development: Expanding Concepts, Vanishing Borders. (Parkway Publishers, Boone, North Carolina): 122-143, 1997

[43] Benjamin Friedlander; Boaz Porat, "*The Modified Yule-Walker Method of ARMA Spectral Estimation",* IEEE Transactions on Aerospace and Electronic Systems, Volume: AES-20, Issue: 2, 1984

[44] 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, 2010

[45] Adel Soroush, Wen Ma, Yule Silvino, Md. Saifur Rahaman, "*Surface modification of thin film composite forward osmosis membrane by silver-decorated graphene-oxide nanosheets*", Environ. Sci.: Nano, 2, 395-405,DOI: 10.1039/C5EN00086F, 2015

[46] Sony Hartono Wijaya, Farit Mochamad Afendi, Irmanida Batubara, Latifah K. Darusman, Md Altaf-Ul-Amin, Shigehiko Kanaya, "*Finding an appropriate equation to measure similarity between binary vectors: Case studies on Indonesian and Japanese herbal medicines*", BMC Bioinformatics BMC series – open, inclusive and trusted 2016 17:520, https://doi.org/10.1186/s12859-016-1392-z, 2016

[47] ASCII of Wikipedia, *https://en.wikipedia.org/wiki/ASCII*, 2017

[48] ASCII Codes Table, *http://ascii.cl/*, 2017

[49] ASCII Table, *http://www.theasciicode.com.ar/*, 2017

[50] ASCII Character Set (2017) http://ee.hawaii.edu/~tep/EE160/Book/chap4/subsection2.1.1.1.html

[51] ASCII Alphabet Characters, *http://www.kerryr.net/pioneers/ascii2.htm*, 2017

[52] ASCII Code, *http://www.ascii-code.net/*, 2017

[53] Decimal to Binary Converter, *http://www.binaryhexconverter.com/decimal-to-binary-converter,* 2017

[54] Decimal to binary, *http://www.rapidtables.com/convert/number/decimal-to-binary.htm,* 2017

[55] Converting from decimal to binary, *https://www.khanacademy.org/math/algebra-home/alg-intro-to-algebra/algebra-alternate-number-bases/v/decimal-to-binary,* 2017

[56] wikiHow to Convert from Decimal to Binarym, *http://www.wikihow.com/Convert-from-Decimal-to-Binary*, 2017

[57] Converting Decimal Numbers to Binary Numbers, *http://interactivepython.org/runestone/static/pythonds/BasicDS/ConvertingDecimalNumberstoBinaryNumbers.html,* 2017

[58] Binary to Decimal Conversion, *http://www.electronics-tutorials.ws/binary/bin_2.html,* 2017

[59] 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, 21-45, 2016

[60] 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, 77-88, 2016

[61] 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), 53-62, New York USA, 2016

[62] 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, 171-179, India, 2016

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

[64] 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

[65] 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.

[66] 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, 1-22, November 2016.

[67] Phu Vo Ngoc, 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.

[68] 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, 31 pages, May 2017.

[69] 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, 23 pages

APPENDICES:

Table 1: Comparisons of our model's results with the works related to [1-32].

No Mention: NM
English Language: EL
YULE'S SIGMA coefficient (YSC)
Semantic classification, sentiment classification: SC
Clustering technique: CT.
Parallel network system: PNS (distributed system).
Special Domain: SD.
Depending on the training data set: DT.
Vector Space Model: VSM

Studies	PMI	JM	Language	SD	DT	YSC	SC	Other measures	Search engines
[1]	Yes	No	English	Yes	Yes	No	Yes	No	No Mention
[2]	Yes	No	English	Yes	No	No	Yes	Latent Semantic Analysis (LSA)	AltaVista
[3]	Yes	No	English	Yes	Yes	No	Yes	Baseline; Turney-inspired; NB; Cluster+NB; Human	AltaVista
[4]	Yes	No	English German	Yes	Yes	No	Yes	SimRank	Google search engine
[5]	Yes	No	English Macedonian	Yes	Yes	No	Yes	No Mention	AltaVista search engine
[6]	Yes	No	English Arabic	Yes	No	No	Yes	No Mention	Google search engine Bing search engine
[7]	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
[8]	Yes	No	English Spanish	Yes	Yes	No	Yes	SO Calculation SVM	Google
[9]	Yes	No	Chinese Tibetan	Yes	Yes	No	Yes	- Feature selection -Expectation Cross Entropy -Information Gain	No Mention
[10]	Yes	No	Chinese	Yes	Yes	No	Yes	DF, CHI, MI andIG	No Mention
[11]	Yes	No	Chinese	Yes	No	No	Yes	Information Bottleneck Method (IB); LE	AltaVista

[12]	Yes	No	Chinese	Yes	Yes	No	Yes	SVM	Google Yahoo Baidu
[13]	Yes	No	Japanese	No	No	No	Yes	Harmonic−Mean	Google and replaced the NEAR operator with the AND operator inthe SO formula.
[14]	Yes	Yes	English	Yes	Yes	No	Yes	Dice; NGD	Google search engine
[15]	Yes	Yes	English	Yes	No	No	Yes	Dice; Overlap	Google
[16]	No	Yes	English	Yes	Yes	No	Yes	A Jaccard index based clustering algorithm (JIBCA)	No Mention
[17]	No	Yes	English	Yes	Yes	No	Yes	Naıve Bayes, Two-Step Multinomial Naıve Bayes, and Two-Step Polynomial-Kernel Support Vector Machine	Google
[18]	No	Yes	Arabic	No	No	No	Yes	Naive Bayes (NB); Support Vector Machines (SVM); Rocchio; Cosine	No Mention
[19]	No	Yes	Chinese	Yes	Yes	No	Yes	A new score–Economic Value (EV), etc.	Chinese search
[20]	No	Yes	Chinese	Yes	Yes	No	Yes	Cosine	No Mention
[21]	No	Yes	English	No	Yes	No	Yes	Cosine	No Mention
[22]	No	Yes	Chinese	No	Yes	No	Yes	Dice; overlap; Cosine	No Mention
[28]	No	No	Vietnamese	No	No	No	Yes	Ochiai Measure	Google
[29]	No	No	English	No	No	No	Yes	Cosine coefficient	Google
[30]	No	No	English	No	No	No	Yes	Sorensen measure	Google
[31]	No	Yes	Vietnamese	No	No	No	Yes	Jaccard	Google
[32]	No	No	English	No	No	No	Yes	Tanimoto coefficient	Google
Our work	No	No	English Language	No	No	Yes	Yes	No	Google search engine

Table 2: Comparisons of our model's advantages and disadvantages with the works related to [1-32].

Surveys	Approach	Advantages	Disadvantages
[1]	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 performance of the developed sentiment lexicons.
[2]	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
[3]	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
[4]	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.
[5]	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 used as seeds.

[6]	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.
[7]	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.
[8]	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
[9]	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
[10]	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
[11]	Adapting Information Bottleneck	The authors' theory verifies the convergence property of the proposed method. The empirical results also support	In this study, only the mutual information measure is employed to measure the three kinds of relationship. In order to show the

	Method for Automatic Construction of Domain-oriented Sentiment Lexicon	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.	robustness of the framework, the authors' future effort is to investigate how to integrate more measures into this framework.
[12]	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
[13]	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.
[14]	In this survey, the authors empirically evaluate the performance of different corpora in sentiment similarity measurement, which is the fundamental task for word polarity classification.	Experiment results show that the Twitter data can achieve a much better performance than the Google, Web1T and Wikipedia based methods.	No Mention
[15]	Adjective-Based Estimation of Short Sentence's Impression	The adjectives are ranked and top n_a 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.
[16]	Jaccard Index based Clustering Algorithm for Mining Online	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	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,

	Review	that can be useful for forecasting sales.	etc.
[17]	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
[18]	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
[19]	Automatic Construction of Financial Semantic Orientation Lexicon from Large-Scale Chinese News Corpus	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 lexicon from large-scale Chinese news corpus automatically ...	No Mention
[20]	Sentiment Classification in Under-Resourced Languages Using Graph-based Semi-supervised Learning Methods	In particular, the authors found that choosing initially labeled vertices in accordance 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
[21]	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
[22]	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	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.

		propagation is promising.	
[28]	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.
[29]	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.
[30]	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
[31]	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	it has a low accuracy.

		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	
[32]	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 binary bits and the sentiment values of the sentiment lexicons to classify the documents of the testing data set to either the positive or the negative in both the sequential environment and the parallel system. -Yule's Sigma coefficient (YSC) through the Google search engine with AND operator and OR operator. -The advantages and disadvantages of this survey are shown in the Conclusion section.		

Table 3: The results of the documents in the testing data set.

	Testing Dataset	**Correct Classification**	**Incorrect Classification**
Negative	4,500,000	4,000,557	499,443
Positive	4,500,000	4,010,343	489,657
Summary	9,000,000	8,010,900	989,100

Table 4: The accuracy of our novel model for the documents in the testing data set.

Proposed Model	Class	Accuracy
Our new model	Negative	89.01%
	Positive	

Table 5: Average time of the classification of our novel model for the documents in testing data set.

	Average time of the classification /9,000,000 documents.
The novel model in the sequential environment	35,091,827 seconds
The novel model in the Cloudera distributed system with 3 nodes	10,363,942 seconds
The novel model in the Cloudera distributed system with 6 nodes	5,981,971 seconds
The novel model in the Cloudera distributed system with 9 nodes	3,921,314 seconds

Table 6: Comparisons of our model's results with the works related to the Yule's Sigma coefficient (YSC) in [42-46].

Studies	PMI	JM	YSC	Language	SD	DT	Sentiment Classification
[42]	Yes	Yes	Yes	English	NM	NM	No mention
[43]	No	No	Yes	NM	NM	NM	No mention
[44]	No	No	Yes	NM	NM	NM	No mention
[45]	No	No	Yes	NM	NM	NM	No mention
[46]	No	No	Yes	NM	NM	NM	No mention
Our work	No	No	Yes	English Language	No	No	Yes

Table 7: Comparisons of our model's benefits and drawbacks with the studies related to the Yule's Sigma coefficient (YSC) in [42-46].

Surveys	Approach	Benefits	Drawbacks
[42]	Assessment of Similarity Indices for Undesirable Properties and a new Tripartite Similarity Index Based on Cost Functions	The purpose of this survey is to motivate, describe, and offer an implementation for, a working similarity index that avoids the difficultiesnoted for the others.	No mention
[43]	The Modified Yule-Walker Method of ARMA Spectral Estimation	The Akaike information criterion is proposed for determining the equation order. A procedure for removing spurious noise modes based on modal decomposition of the sample covariance matrix is derived. The role of the singular value decomposition method in solving the modified Yule-Walker equations is discussed. A number of techniques for estimating MA spectral parameters are presented.	No mention
[44]	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 similarityand distance measures used over the last century and reveal their correlations through the hierarchical clustering technique	No mention
[45]	Surface modification of thin film composite forward osmosis membrane by silver-decorated graphene-oxide nanosheets	In this study, silver nanoparticle (AgNPs)-decorated graphene oxide (GO) nanosheets (as an effective biocidal material) were covalently bonded to the PA surface to impart improved hydrophilicity and antibacterial properties to the membrane. AgNPs were synthesized *in situ* by the wet chemical reduction of silver nitrate onto the surface of GO nanosheets. The formation of the composite was verified by UV-vis spectroscopy, X-ray diffraction, and transmission electron microscopy techniques. The synthesized GO/Ag nanocomposites were then covalently bonded onto the TFC PA membrane surface using cysteamine through an amide forming condensation reaction. ATR-FTIR and XPS results confirmed the covalent bonding of the nanocomposite onto the TFC PA surface. Overall, the GO/Ag nanocomposite functionalized	No mention

		membranes exhibited super-hydrophilic properties (contact angles below 25°) and significant bacterial (*E. coli*) inactivation (over 95% in static bacterial inactivation tests) without adversely affecting the membrane transport properties.	
[46]	Finding an appropriate equation to measure similarity between binary vectors: case studies on Indonesian and Japanese herbal medicines	The selection of binary similarity and dissimilarity measures for multivariate analysis is data dependent. The proposed method can be used to find the most suitable binary similarity and dissimilarity equation wisely for a particular data. The authors' finding suggests that all four types of matching quantities in the Operational Taxonomic Unit (OTU) table are important to calculate the similarity and dissimilarity coefficients between herbal medicine formulas. Also, the binary similarity and dissimilarity measures that include the negative match quantity *d* achieve better capability to separate herbal medicine pairs compared to equations that exclude *d*	No mention
Our work	-We use the binary bits and the sentiment values of the sentiment lexicons to classify the documents of the testing data set to either the positive or the negative in both the sequential environment and the parallel system. -Yule's Sigma coefficient (YSC) through the Google search engine with AND operator and OR operator. -The advantages and disadvantages of this survey are shown in the Conclusion section.		

Table 8: Comparisons of our model with the latest sentiment classification models (or the latest sentiment classification methods) in [59-69]

Studies	CT	Sentiment Classification	PNS	SD	DT	Language
[59]	No	Yes	NM	Yes	Yes	Yes
[60]	No	Yes	NM	Yes	Yes	NM
[61]	No	Yes	NM	Yes	Yes	EL
[62]	No	Yes	NM	Yes	Yes	NM
[63]	No	Yes	No	No	No	EL
[64]	No	Yes	No	No	No	EL
Our work	Yes	Yes	Yes	No	No	Yes

Table 9: Comparisons of our model's positives and negatives the latest sentiment classification models (or the latest sentiment classification methods) in [59-69]

Studies	Approach	Positives	Negatives
[59]	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
[60]	Semantic	This approach initially mines sentiment-bearing terms from the	No mention

	Orientation-Based Approach for Sentiment Analysis	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.	
[61]	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 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.	A line of future research would be to explore the authors' meta features with other classification algorithms and feature selection techniques in different sentiment analysis tasks such as scoring movies or products according to their related reviews.
[62]	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
[63]	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 Database data set.	No mention
[64]	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 binary bits and the sentiment values of the sentiment lexicons to classify the documents of the testing data set to either the positive or the negative in both the sequential environment and the parallel system. -Yule's Sigma coefficient (YSC) through the Google search engine with AND operator and OR operator. -The positives and negatives of the proposed model are given in the Conclusion section.		