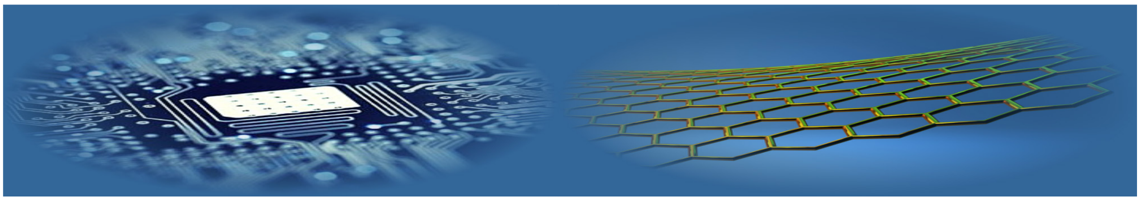

English sentiment classification using a BIRCH algorithm and the sentiment lexicons-based one-dimensional vectors in a parallel network environment

Vo Ngoc Phu[1]*, Vo Thi Ngoc Tran[2]

[1]Nguyen Tat Thanh University, 300A Nguyen Tat Thanh Street, Ward 13, District 4, Ho Chi Minh City, 702000, Vietnam
[2]School of Industrial Management (SIM), Ho Chi Minh City University of Technology - HCMUT, Vietnam National University, Ho Chi Minh City, Vietnam

**Corresponding author e-mail: vongocphu03hca@gmail.com; vongocphu@ntt.edu.vn*

Received 9 February 2018, www.cmnt.lv

Abstract

Sentiment classification is significant in everyday life, such as in political activities, commodity production, and commercial activities. In this survey, we have proposed a new model for Big Data sentiment classification. We use a Balanced Inerative Reducing and Clustering using Hierarchies algorithm (BIRCH) and many one-dimensional vectors based on many sentiment lexicons of our basis English sentiment dictionary (bESD) to cluster one document of our testing data set based on our training data set in English. We calculate the sentiment scores of English terms (verbs, nouns, adjectives, adverbs, etc.) by using a BARONI-URBANI & BUSER-II coefficient (BUBC) through a Google search engine with AND operator and OR operator. We do not use any multi-dimensional vector. We also do not use any one-dimensional vector based on a vector space modelling (VSM). We do not use any similarity coefficient of a data mining field. The BIRCH is used in clustering one sentence of one document of the testing data set into either the positive or the negative of the training data set. We tested the proposed model in both a sequential environment and a distributed network system. We achieved 87.76% accuracy of the testing data set. The execution time of the model in the parallel network environment is faster than the execution time of the model in the sequential system. The results of this work can be widely used in applications and research of the English sentiment classification.

Key words

computer and information technologies,
natural and engineering sciences,
operation research and decision making,
mathematical and computer modelling

1 Introduction

Sentiment analysis is implemented by using machine-learning methods or methods based on lexicons; or a combination of both. Opinion mining relates to emotional researches that are expressed in documents. Sentiment analysis has a wide range of applications in the fields of business, organizations, governments and individuals.

Clustering data is to process a set of objects into classes of similar objects. One cluster is a set of data objects which are similar to each other and are not similar to objects in other clusters. A number of data clusters can be clustered, which can be identified following experience or can be automatically identified as part of clustering method.

To implement our new model, we propose the following basic principles:

- Assuming that each English sentence has m English words (or English phrases).
- Assuming that the maximum number of one English sentence is m_max; it means that m is less than m_max or m is equal to m_max.
- Each English sentence is transferred into one vector (one-dimensional). Thus, the length of the vector is m. If m is less than m_max then each element of the vector from m to m_max-1 is 0 (zero).
- All the sentences of one document of the testing data set are transferred into the one-dimensional vectors of one document of the testing data set based on many sentiment lexicons of our basis English sentiment dictionary (bESD).
- All the positive sentences of the training data set are transferred the positive one-dimensional vectors based on the sentiment lexicons of the bESD, called the positive vector group of the training data set.
- All the negative sentences of the training data set are transferred the negative one-dimensional vectors based on the sentiment lexicons of the bESD, called the negative vector group of the training data set.

The aim of this survey is to find a new approach to improve the accuracy of the sentiment classification results and to shorten the execution time of the proposed model with a low cost.

The motivation of this new model is as follows: Many Algorithm in the data mining field can be applied to natural language processing, specifically semantic classification for processing millions of English documents. A BARONI-URBANI & BUSER-II similarity measure (BUBC) and a Balanced Inerative Reducing and Clustering using Hierarchies algorithm (BIRCH) of the clustering technologies of the data mining filed can be applied to the sentiment classification in both a sequential environment and a parallel network system. This will result in many discoveries in scientific research, hence the motivation for this study.

The novelty of the proposed approach is that the BARONI-URBANI & BUSER-II similarity measure (BUBC) and the BIRCH is applied to sentiment analysis. This algorithm can also be applied to identify the emotions of millions of documents. This survey can be applied to other parallel network systems. Hadoop Map (M) and Hadoop Reduce (R) are used in the proposed model. Therefore, we will study this model in more detail.

To get higher accuracy of the results of the sentiment classification and shorten execution time of the sentiment classification, We use many sentiment lexicons in English of our basis English sentiment dictionary (bESD). We do not use any multi-dimensional vector based on both VSM [51-53] and the sentiment lexicons of the bESD. We also do not use any one-dimensional vector based on a vector space modelling VSM [51-53]. We do not use any similarity coefficient of a data mining field. We only use many one-dimensional vectors based on the sentiment lexicons of the bESD. We identify the sentiment scores of English terms (verbs, nouns, adjectives, adverbs, etc.) of the bESD by using a BARONI-URBANI & BUSER-II coefficient (BUBC) through the Google search engine with AND operator and OR operator. All the sentences of one document of the testing data set are transferred into the one-dimensional vectors of one document of the testing data set based on the sentiment lexicons of our basis English sentiment dictionary. All the positive sentences of the training data set are transferred the positive one-dimensional vectors based on the sentiment lexicons of the bESD, called the positive vector group of the training data set. All the negative sentences of the training data set are transferred the negative one-dimensional vectors based on the sentiment lexicons of the bESD, called the negative vector group of the training data set. Then, we use the BIRCH to cluster one one-dimensional vector (corresponding to one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set. This one-dimensional vector is the positive polarity if it is clustered into the positive vector group. The vector is the negative if it is clustered into the negative vector group. The vector is neutral polarity if it is not clustered into both the positive vector group and the negative vector group. One document of the testing data set is the positive if the number of one-dimensional vectors clustered into the positive is greater than that clustered into the negative. One document of the testing data set is the negative if the number of one-dimensional vectors clustered into the positive is less than that clustered into the negative. One document of the testing data set is the neutral if the number of one-dimensional vectors clustered into the positive is as equal as that clustered into the negative.

We perform all the above things in the sequential system firstly. To shorten execution time of the proposed model, we implement all the above things in the distributed environment secondly.

Our model has many significant applications to many areas of research as well as commercial applications:

1. Many surveys and commercial applications can use the results of this work in a significant way.
2. The Algorithm 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 BIRCH – related Algorithm are proposed in this survey.
17. The BUBC – related Algorithm are built in this work.

This study contains 6 sections. Section 1 introduces the study; Section 2 discusses the related works about the vector space modelling (VSM), BARONI-URBANI & BUSER-II similarity measure (BUBC), Balanced Inerative Reducing and Clustering using Hierarchies algorithm (M), 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 (Poinwise 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 BARONI-URBANI & BUSER-II coefficient (BUBC) is also used for identifying valence and polarity of one English word (or one English phrase). Finally, we identify the sentimental values of English verb phrases based on the basis English semantic lexicons of the basis English emotional dictionary (bESD).

There are the works related to PMI measure in [1-13]. In the research [1], the authors generate several Norwegian sentiment lexicons by extracting sentiment information from two different types of Norwegian text corpus, namely, news corpus and discussion forums. The methodology is based on the Poin wise Mutual Information (PMI). The authors introduce a modification of the PMI that considers small "blocks" of the text instead of the text as a whole. The study in [2] introduces a simple algorithm for unsupervised learning of semantic orientation from extremely large corpora, etc.

Two studies related to the PMI measure and Jaccard measure are in [14, 15]. In the survey [14], the authors empirically evaluate the performance of different corpora in sentiment similarity measurement, which is the fundamental task for word polarity classification. The research in [15] proposes a new method to estimate impression of short sentences considering adjectives. In the proposed system, first, an input sentence is analysed and pre-processed to obtain keywords. Next, adjectives are taken out from the data which is queried from Google N-gram corpus using keywords-based templates.

The works related to the Jaccard measure are in [16-22]. The survey in [16] investigates the problem of sentiment analysis of the online review. In the study [17], the authors are addressing the issue of spreading public concern about epidemics. Public concern about a communicable disease can be seen as a problem of its own, etc.

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

The English dictionaries are [33-38] and there are more than 55,000 English words (including English nouns, English adjectives, English verbs, etc.) from them.

The studies related the Balanced Interative Reducing and Clustering using Hierarchies algorithm (BIRCH) are in [39-44]. The authors in [39] evaluate BIRCH'S time/space efficiency, data input order sensitivity, and clustering quality through several experiments. In this study [40], an efficient and scalable data clustering method is proposed, based on a new in-memory data structure called CF-tree, which serves as an in-memory summary of the data distribution. The authors have implemented it in a system called BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), and studied its performance extensively in terms of memory requirements, running time, clustering quality, stability and scalability; the authors also compare it with other available methods, etc.

There are the works related to the BARONI-URBANI & BUSER-II coefficient (BUBC) in [45-50]. The authors in [50] collected 76 binary similarity and distance measures used over the last century and reveal their correlations through the hierarchical clustering technique, etc.

There are the works related to vector space modelling (VSM) in [51-53]. In this study [51], the authors examined the Vector Space Model, an Information Retrieval technique and its variation. In this survey [52], the authors consider multi-label text classification task and apply various feature sets. The authors consider a subset of multi-labeled files from the Reuters-21578 corpus. The authors use traditional tf-IDF values of the features and tried both considering and ignoring stop words. The authors also tried several combinations of features, like bigrams and unigrams. The authors in [53] introduce a new-weighting method based on statistical estimation of the importance of a word for a specific categorization problem. This method also has the benefit to make feature selection implicit, since useless features for the categorization proble considered to get a very small weight.

The latest researches of the sentiment classification are [54-64]. In the research [54], the authors present their machine learning experiments with regard to sentiment analysis in blog, review and forum texts found on the World Wide Web and written in English, Dutch and French. The survey in [55] discusses an approach where an exposed stream of tweets from the Twitter micro blogging site are pre-processed and classified based on their sentiments. In sentiment classification system the concept of opinion subjectivity has been accounted. In the study, the authors present opinion detection and organization subsystem, which have already been integrated into our larger question-answering system, etc.

3 Data set

In Fig 1 below, the testing data set includes 6,500,000,000 documents in the movie field, which contains 3,250,000,000 positive documents and 3,250,000,000 negative documents in English. All the documents in our testing data set are automatically extracted from English Facebook, English websites and social networks; then we labelled positive and negative for them.

In Fig 2 below, the training data set

includes 4,000,000,000 sentences in the movie field, which contains 2,000,000,000 positive sentences and 2,000,000,000 negative sentences in English. All the sentences in our training data set are automatically extracted from English Facebook, English websites and social networks; then we labelled positive and negative for them.

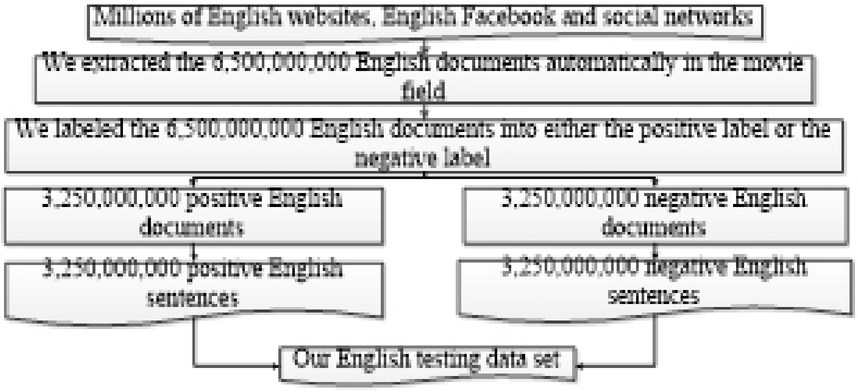

FIGURE 1 Our English testing data set

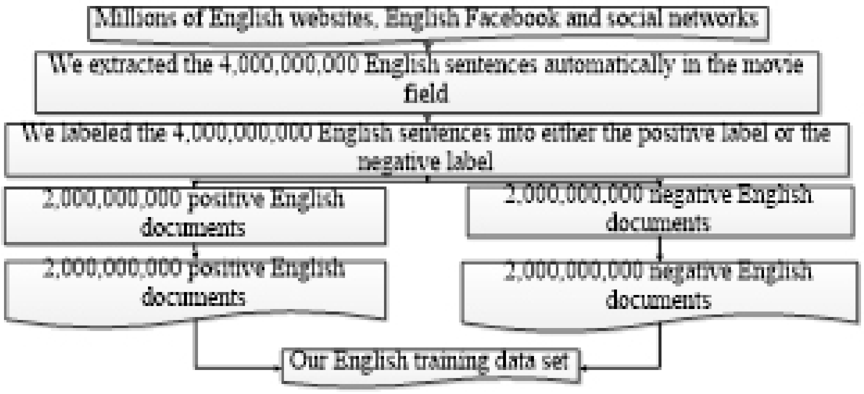

FIGURE 2 Our English training data set

4 Methodology

This section comprises two parts. The first part is to create the sentiment lexicons in English in both a sequential environment and a distributed system in the sub-section (4.1). The second part is to use the BIRCH and the one-dimensional vectors to classify the documents of the testing data set into either the positive or the negative in both a sequential environment and a distributed system in the sub-section (4.2).

In the sub-section (4.1), the section includes three parts. The first sub-section of this section is to identify a sentiment value of one word (or one phrase) in English in the sub-section (4.1.1). The second part of this section is to create a basis English sentiment dictionary (bESD) in a sequential system in the sub-section (4.1.2). The third sub-section of this section is to create a basis English sentiment dictionary (bESD) in a parallel environment in the sub-section (4.1.3).

In the sub-section (4.2), the section compises two parts. The first part of this section is to use the BIRCH and the one-dimensional vectors to classify the documents of the testing data set into either the positive or the negative in a sequential environment in the sub-section (4.2.1). The second part of this section is to use the BIRCH and the one-dimensional vectors to classify the documents of the testing data set into either the positive or the negative in the parallel network environment in the sub-section (4.2.2).

4.1 CREATING THE SENTIMENT LEXICONS IN ENGLISH

The section includes three parts. The first sub-section of this section is to identify a sentiment value of one word (or one phrase) in English in the sub-section (4.1.1). The second part of this section is to create a basis English sentiment dictionary (bESD) in a sequential system in the sub-section (4.1.2). The third sub-section of this section is to create a basis English sentiment dictionary (bESD) in a parallel environment in the sub-section (4.1.3).

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

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

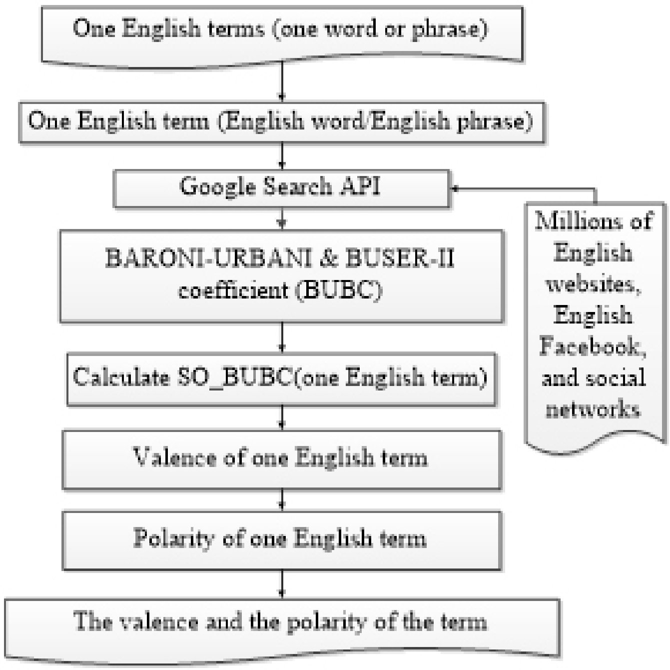

FIGURE 2 Our English training data set

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

$$PMI(w_i, w_j) = \log_2 \left(\frac{P(w_i, w_j)}{P(w_i) \times P(w_j)} \right) \quad (1)$$

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

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

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 Chinese, 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.

According to [14-22], Jaccard between two words wi and wj has the equations

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

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

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

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

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

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 Vietnamese. 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 BARONI-URBANI & BUSER-II coefficient (BUBC) are the similarity measures between two words, and they can perform the same functions and with the same characteristics; so BUBC is used in calculating the valence of the words. In addition, we prove that BUBC can be used in identifying the valence of the English word through the Google search with the AND operator and OR operator.

With the BARONI-URBANI & BUSER-II coefficient (BUBC) in [45, 50], we have the equation of the BUBC (with a and b are the vectors):

$$BARONI-URBANI\,\&\,BUSER-IICoefficient(a,b) = BARONI-URBANI\,\&\,BUSER-IIMeasure(a,b)$$
$$= BUBC(a,b) - \frac{\sqrt{(a\cap b)*(\neg a\cap\neg b)} + (a\cap b) - [(\neg a\cap b)+(a\cap\neg b)]}{\sqrt{(a\cap b)*(\neg a\cap\neg b)} + (a\cap b) + (\neg a\cap b) + (a\cap\neg b)} \quad (6)$$

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

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

$$BARONI-URBANI\,\&\,BUSER-IIMeasure(w_1,w_2) = BARONI-URBANI\,\&\,BUSER-IIcoefficient(w_1,w_2) =$$
$$BUBC(w_1,w_2) = \frac{\sqrt{P(w_1,w_2)*P(\neg w_1,\neg w_2)} + P(w_1,w_2) - [P(\neg w_1,w_2)+P(w_1,\neg w_2)]}{\sqrt{P(w_1,w_2)*P(\neg w_1,\neg w_2)} + P(w_1,w_2) + P(\neg w_1,w_2) + P(w_1,\neg w_2)} \quad (7)$$

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

$$Valence(w) = SO\_BUBC(w) = BUBC(w, positive\_query) - BUBC(w, negative\_query) \quad (8)$$

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

$$BUBC(w, positive\_query) = \frac{A9 + P(w, positive\_query) - B9}{A9 + P(w, positive\_query) + B9} \quad (9)$$

With

$$A9 = \sqrt{P\left(w, positive\_query\right) * P\left(\neg w, \neg positive\_query\right)}$$

$$B9 = P\left(\neg w, positive\_query\right) + P\left(w, \neg positive\_query\right)$$

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

$$BUBC\left(w, negative\_query\right) = \frac{A10 + P\left(w, negative\_query\right) - B10}{A10 + P\left(w, negative\_query\right) + B10} \quad (10)$$

with

$$A10 = \sqrt{P\left(w, negative\_query\right) * P\left(\neg w, \neg negative\_query\right)}$$

$$B10 = P\left(\neg w, negative\_query\right) + P\left(w, \neg negative\_query\right)$$

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_BUBC(w): valence of English word (or English phrase) w; is SO of word (or phrase) by using the BARONI-URBANI & BUSER-II coefficient (BUBC)
6. positive_query: { active or good or positive or beautiful or strong or nice or excellent or fortunate or correct or superior } with the positive query is the a group of the positive English words.
7. negative_query: { passive or bad or negative or ugly or week or nasty or poor or unfortunate or wrong or inferior } with the negative_query is the a group of the negative English words.
8. P(w, positive_query): number of returned results in Google search by keyword (positive_query and w). We use the Google Search API to get the number of returned results in search online Google by keyword (positive_query and w)
9. P(w, negative_query): number of returned results in Google search by keyword (negative_query and w). We use the Google Search API to get the number of returned results in search online Google by keyword (negative_query and w)
10. P(w): number of returned results in Google search by keyword w. We use the Google Search API to get the number of returned results in search online Google by keyword w
11. P(¬w,positive_query): number of returned results in Google search by keyword ((not w) and positive_query). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and positive_query).
12. P(w, ¬positive_query): number of returned results in the Google search by keyword (w and (not (positive_query))). We use the Google Search API to get the number of returned results in search online Google by keyword (w and [not (positive_query)]).
13. P(¬w, ¬positive_query): number of returned results in the Google search by keyword (w and (not (positive_query))). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and [not (positive_query)]).
14. P(¬w,negative_query): number of returned results in Google search by keyword ((not w) and negative_query). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and negative_query).
15. P(w,¬negative_query): number of returned results in the Google search by keyword (w and (not (negative_query))). We use the Google Search API to get the number of returned results in search online Google by keyword (w and (not (negative_query))).
16. P(¬w,¬negative_query): number of

returned results in the Google search by keyword (w and (not (negative_query))). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and (not (negative_query))).

As like Cosine, Ochiai, Sorensen, Tanimoto, PMI and Jaccard about calculating the valence (score) of the word, we identify the valence (score) of the English word w based on both the proximity of positive_query with w and the remote of positive_query with w; and the proximity of negative_query with w and the remote of negative_query with w.The English word w is the nearest of positive_query if BUBC (w, positive_query) is as equal as 1.The English word w is the farthest of positive_query if BUBC(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 BUBC(w, positive_query) > 0 and BUBC(w, positive_query) ≤ 1.The English word w is the nearest of negative_query if BUBC(w, negative_query) is as equal as 1. The English word w is the farthest of negative_query if BUBC(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 BUBC(w, negative_query) > 0 and BUBC(w, negative_query) ≤ 1. So, the valence of the English word w is the value of BUBC(w, positive_query) substracting the value of BUBC(w, negative_query) and the eq. (8) is the equation of identifying the valence of the English word w.

We have the information about BUBC as follows:

1. BUBC(w, positive_query) ≥ 0 and BUBC(w, positive_query) ≤ 1.
2. BUBC(w, negative_query) ≥ 0 and BUBC (w, negative_query) ≤ 1
3. If BUBC (w, positive_query) = 0 and BUBC (w, negative_query) = 0 then SO_BUBC (w) = 0.
4. If BUBC (w, positive_query) = 1 and BUBC (w, negative_query) = 0 then SO_BUBC (w) = 0.
5. If BUBC (w, positive_query) = 0 and BUBC (w, negative_query) = 1 then SO_BUBC (w) = -1.
6. If BUBC (w, positive_query) = 1 and BUBC (w, negative_query) = 1 then SO_BUBC(w) = 0.

So, SO_BUBC (w) ≥ -1 and SO_BUBC (w) ≤ 1.

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

We calculate the valence and the polarity of the English word or phrase w using a training corpus of approximately one hundred billion English words — the subset of the English Web that is indexed by the Google search engine on the internet. AltaVista was chosen because it has a NEAR operator. The AltaVista NEAR operator limits the search to documents that contain the words within ten words of one another, in either order. We use the Google search engine which does not have a NEAR operator; but the Google search engine can use the AND operator and the OR operator. The result of calculating the valence w (English word) is similar to the result of calculating valence w by using AltaVista. However, AltaVista is no longer.

In summary, by using eq. (8), eq. (9), and eq. (10), we identify the valence and the polarity of one word (or one phrase) in English by using the SC through the Google search engine with AND operator and OR operator.

In Table 1 and Table 2 below of the Appendices section, we compare our model's results with the surveys in [1- 22].

In Table 3 and Table 4 below, we compare our model's results with the researches related to the BARONI-URBANI & BUSER-II coefficient(BUBC) in [39,40].

4.1.2 Creating a basis English sentiment dictionary (bESD) in a 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 BUBC in a sequential system, as the following diagram Figure 4 below shows.

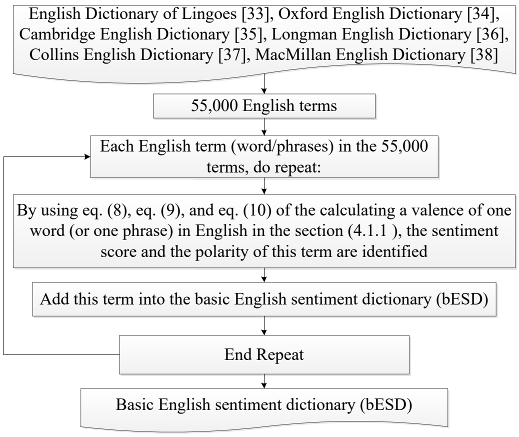

FIGURE 4 Overview of creating a basis English sentiment dictionary (bESD) in a sequential environment

We proposed the algorithm 1 to perform this section. The main ideas of the algorithm 1 are as follows:

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

Output: a basis English sentiment dictionary (bESD)

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

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

Step 3: Add this term into the basis English sentiment dictionary (bESD);

Step 4: End Repeat – End Step 1;

Step 5: Return bESD;

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

4.1.3 Creating a basis English sentiment dictionary (bESD) in a 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 BUBC in a parallel network environment, as the following diagram Figure 5 below shows.

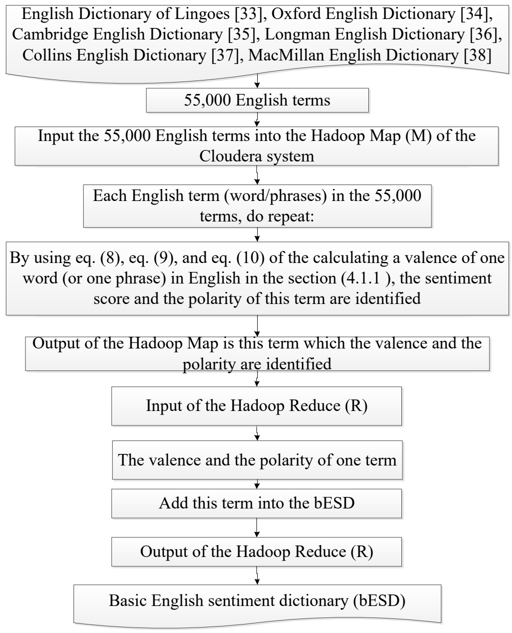

FIGURE 5 Overview of creating a basis English sentiment dictionary (bESD) in a distributed environment

Figure 5, this section includes two phases: the Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase is the 55,000 terms in English in [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 propose the algorithm 2 to implement the Hadoop Map phase. The main ideas of the algorithm 2 are as follows:

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

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

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

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

Step 3: Return this term;

We propose the algorithm 3 to perform the Hadoop Reduce phase. The main ideas of the algorithm 3 are as follows:

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

Output: a basis English sentiment dictionary (bESD)

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

Step 2: Return bESD;

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

4.2 USING THE BIRCH AND THE ONE-DIMENSIONAL VECTORS TO CLASSIFY THE DOCUMENTS OF THE TESTING DATA SET INTO EITHER POLARITY OR THE NEGATIVE POLARITY

This section comprises two parts as follows: The first part of this section is to use the BIRCH and the one-dimensional vectors to classify the documents of the testing data set into either the positive polarity or the negative polarity in a sequential environment in the sub-section (4.2.1). The second part of this section is to use the BIRCH and the one-dimensional vectors to classify the documents of the testing data set into either the positive polarity or the negative polarity in a distributed system in the sub-section (4.2.2).

4.2.1 Using the BIRCH and the one-dimensional vectors to classify the documents of the testing data set into either polarity or the negative polarity in the sequential environment

Figure 6, we use the BIRCH and the one-dimensional vectors to classify the documents of the testing data set into either polarity or the negative polarity in the sequential environment as follows:

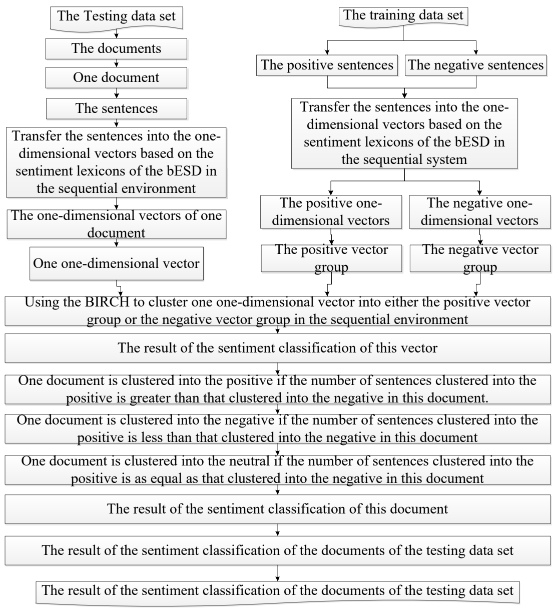

FIGURE 6 Overview of using the BIRCH and the one-dimensional vectors to classify the documents of the testing data set into either polarity or the negative polarity in the sequential environment

Figure 6, we perform the proposed model in the sequential system as follows: firstly, we create the sentiment lexicons of the bESD based on the creating a basis English sentiment dictionary (bESD) in a sequential environment in (4.1.2). We transfer one sentence into one one-dimensional vector based on the sentiment lexicons of the bESD. We transfer all the sentences of one document of the testing data set into the one-dimensional vectors based on the sentiment lexicons of the bESD. All the positive sentences of the training data set are transferred into the positive one-dimensional vectors based on the sentiment lexicons of the bESD, called the positive vector group of the training data set. All the negative sentences of the training data set are transferred into the negative one-dimensional vectors based on the sentiment lexicons of the bESD, called the negative vector group of the training data set. Then, we use the BIRCH to cluster one one-dimensional vector (corresponding to one sentence of one document of the testing data set) into either the positive vector group or the negative vector group. One document is clustered into the positive if the number of sentences clustered into the positive is greater than that clustered into the negative in this document. One document is clustered into the negative if the number of sentences 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 sentences clustered into the positive is as equal as that clustered into the negative in this document. Finally, all the documents of the testing data set are clustered into either the positive or the negative.

We propose the algorithm 4 to transfer one sentence into one-dimensional vector based on the sentiment lexicons of the bESD in the sequential environment. The main ideas of the algorithm 4 are as follows:

Input: one sentence and the bESD

Output: one one-dimensional vector based on the sentiment lexicons of the bESD

Step 1: Split this sentence into the meaningful terms based on the bESD;

Step 2: Set OneOne-dimensionalVector := null;

Step 3: Each term in the terms of this sentence, do repeat:

Step 4: Identify the valence of this term basd on bESD;

Step 5: Add this term into OneOne-dimensionalVector;

Step 6: End Repeat – End Step 3;

Step 7: Return OneOne-dimensionalVector;

We propose the algorithm 5 to transfer all the sentences of one document into the one-dimensional vectors based on the sentiment lexicons of the bESD in the sequential system. The main ideas of the algorithm 5 are as follows:

Input: one document and the bESD;

Output: the one-dimensional vectors of this document;

Step 1: Split this document into the sentences;

Step 2: Each sentence in the sentences of this document, do repeat:

Step 3: OneOne-dimensionalVector := The algorithm 4 to transfer one sentence into one-dimensional vector based on the sentiment lexicons of the bESD in the sequential environment with the input is this sentence and the bESD;

Step 4: Add OneOne-dimensionalVector into the one-dimensional vectors of this document;

Step 5: End Repeat – End Step 3;

Step 6: Return the one-dimensional vectors of this document;

We propose the algorithm 6 to transfer all the positive sentences of the training data set into the one-dimensional vector based on the sentiment lexicons of the bESD in the sequential system, called the positive vector group of the training data set. The main ideas of the algorithm 6 are as follows:

Input: the positive sentences of the training data set and the bESD;

Output: the positive one-dimensional vectors, called the positive vector group of the training data set;

Step 1: Set the positive vector group := null;

Step 2: Each sentence in the positive sentences of the training data set, do repeat:

Step 3: OneOne-dimensionalVector := The algorithm 4 to transfer one sentence into one-dimensional vector based on the sentiment lexicons of the bESD in the sequential environment with the input is this sentence and the bESD;

Step 4: Add dimensionalVector into the positive vector group;

Step 5: End Repeat – End Step 2;

Step 6: Return the positive vector group;

We propose the algorithm 7 to transfer all the negative sentences of the training data set into the one-dimensional vector based on the sentiment lexicons of the bESD in the sequential system, called the negative vector group of the training data set. The main ideas of the algorithm 7 are as follows:

Input: the negative sentences of the training data set and the bESD;

Output: the negative one-dimensional vectors, called the negative vector group of the training data set;

Step 1: Set the negative vector group := null;

Step 2: Each sentence in the negative sentences of the training data set, do repeat:

Step 3: OneOne-dimensionalVector := The algorithm 4 to transfer one sentence into one-dimensional vector based on the sentiment lexicons of the bESD in the sequential environment with the input is this sentence and the bESD;

Step 4: Add dimensionalVector into the negative vector group;

Step 5: End Repeat – End Step 2;

Step 6: Return the negative vector group;

According to the surveys related the Balanced Inerative Reducing and Clustering using Hierarchies algorithm (BIRCH) in [39-44], we propose the algorithm 8 to use the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the sequential environment as follows:

Input: one one-dimensional vector of a document in the testing data set; the positive vector group and the negative vector group of the training data set.

Output: the result of clustering the

vector into either the positive vector group or the negative vector group.

Step 1: Scan all data and build an initial in-memory CF tree, using the given amount of memory and recycling space on disk.

Step 2: With each vector in n vectors, do:

Step 3: Condense into desirable length by building a smaller CF tree.

Step 4: Global clustering with the vector into CF Triple 1 or CF Triple 2

Step 5: Cluster refining – this is optional, and requires more passes over the data to refine the results.

Step 6: Return the result of clustering the vector into either the positive vector group or the negative vector group.

We propose the algorithm 9 to cluster one document of the testing data set into either the positive or the negative in the sequential system. The main ideas of the algorithm 9 are as follows:

Input: one document of the testing data set; the positive vector group and the negative vector group of the training data set.

Output: The result of the sentiment classification of this document

Step 1: TheOne-dimensionalVectors := The algorithm 5 to transfer all the sentences of one document into the one-dimensional vectors based on the sentiment lexicons of the bESD in the sequential system with the input is this document;

Step 2: Set count_positive := 0; and count_negative := 0;

Step 3: Each one-dimensional vector in TheOne-dimensionalVectors, do repeat:

Step 4: OneResult := The algorithm 8 to use the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the sequential environment with this vector, the positive vector group and the negative vector group;

Step 5: If OneResults is the positive Then count_positive := count_positive + 1;

Step 6: Else If OneResults is the negative Then count_negative := count_negative + 1;

Step 7: End Repeat – End Step 3;

Step 8: If count_positive is greater than count_negative Then Return positive;

Step 9: Else If count_positive is less than count_negative Then Return negative;

Step 10: Return neutral;

We build the algorithm 10 to cluster the documents of the testing data set into either the positive or the negative in the sequential environment. The main ideas of the algorithm 10 are as follows:

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

Output: the results of the sentiment classification of the documents of the testing data set;

Step 1: The algorithm 6 to transfer all the positive sentences of the training data set into the one-dimensional vector based on the sentiment lexicons of the bESD in the sequential system, called the positive vector group of the training data set with the input is the positive sentences of the training data set; and the bESD;

Step 2: The algorithm 7 to transfer all the negative sentences of the training data set into the one-dimensional vector based on the sentiment lexicons of the bESD in the sequential system, called the negative vector group of the training data set with the input is the negative sentences of the training data set; and the bESD;

Step 3: Each document in the documents of the testing data set, do repeat:

Step 4: OneResult := the algorithm 9 to cluster one document of the testing data set into either the positive or the negative in the sequential system with the input is this document, the positive vector group and the negative vector group;

Step 5: Add OneResult into the results of the sentiment classification of the documents of the testing data set;

Step 6: Return the results of the sentiment classification of the documents of the testing data set.

4.2.2 Using the BIRCH and the one-dimensional vectors to classify the documents of the testing data set into either polarity or the negative polarity in the distributed network system

Figure 7, we use the BIRCH and the one-dimensional vectors to classify the documents of the testing data set into either polarity or the negative polarity in the sequential environment as follows:

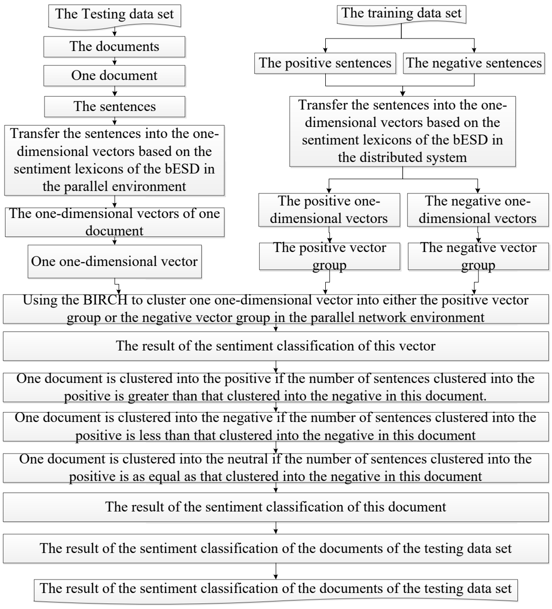

FIGURE 7 Overview of using the BIRCH and the one-dimensional vectors to classify the documents of the testing data set into either polarity or the negative polarity in the distributed network environment

Figure 7, we perform the proposed model in the parallel system as follows: firstly, we create the sentiment lexicons of the bESD based on the creating a basis English sentiment dictionary (bESD) in a distributed system in (4.1.3). We transfer one sentence into one one-dimensional vector based on the sentiment lexicons of the bESD. We transfer all the sentences of one document of the testing data set into the one-dimensional vectors based on the sentiment lexicons of the bESD. All the positive sentences of the training data set are transferred into the positive one-dimensional vectors based on the sentiment lexicons of the bESD, called the positive vector group of the training data set. All the negative sentences of the training data set are transferred into the negative one-dimensional vectors based on the sentiment lexicons of the bESD, called the negative vector group of the training data set. Then, we use the BIRCH to cluster one one-dimensional vector (corresponding to one sentence of one document of the testing data set) into either the positive vector group or the negative vector group. One document is clustered into the positive if the number of sentences clustered into the positive is greater than that clustered into the negative in this document. One document is clustered into the negative if the number of sentences 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 sentences clustered into the positive is as equal as that clustered into the negative in this document. Finally, all the documents of the testing data set are clustered into either the positive or the negative.

Figure 8, we transform one English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera. This stage includes two phases: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map phase is one sentence and the bESD. The output of the Hadoop Map phase is one term (one meaningful word/or one meaningful phrase) which the valence is identified. The input of the Hadoop Reduce phase is the output of the Hadoop Map, thus, the input of the Hadoop Reduce phase is one term (one meaningful word/or one meaningful phrase) which the valence is identified. The output of the Hadoop Reduce phase is one one-dimensional vector of this sentence.

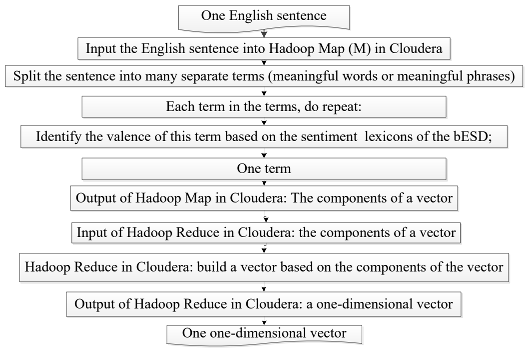

FIGURE 8 Overview of transforming each English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera

We build the algorithm 11 to perform the Hadoop Map phase of transforming each English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera. The main ideas of the algorithm 11 are as follows:

Input: one sentence and the bESD;

Output: one term (one meaningful word/or one meaningful phrase) which the valence is identified

Step 1: Input this sentence and the bESD into the Hadoop Map in the Cloudera system;

Step 2: Split this sentence into the many meaningful terms (meaningful words/or meaningful phrases) based on the bESD;

Step 3: Each term in the terms, do repeat:

Step 4: Identify the valence of this term based on the bESD;

Step 5: Return this term; //the output of the Hadoop Map phase.

We propose the algorithm 12 to perform the Hadoop Reduce phase of transforming each English sentence into one one-dimensionalvector based on the sentiment lexicons of the bESD in Cloudera. The main ideas of the algorithm 12 are as follows:

Input: one term (one meaningful word/or one meaningful phrase) which the valence is identified – the output of the Hadoop Map phase

Output: one one-dimensionalvector based on the sentiment lexicons of the bESD

Step 1: Receive one term;

Step 2: Add this term into the one-dimensionalvector;

Step 3: Return the one-dimensionalvector;

Figure 9, we transfer all the sentences of one document of the testing data set into the one-dimensional vectors of the document of testing data set based on the sentiment lexicons of the bESD in the parallel network environment. This stage comprises two phases: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map is one document of the testing data se. The output of the Hadoop Reduce is one one-dimensional vector (corresponding to one sentence) of this document. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is one one-dimensional vector (corresponding to one sentence) of this document. The output of the Hadoop Reduce is the one-dimensionalvectors of this document.

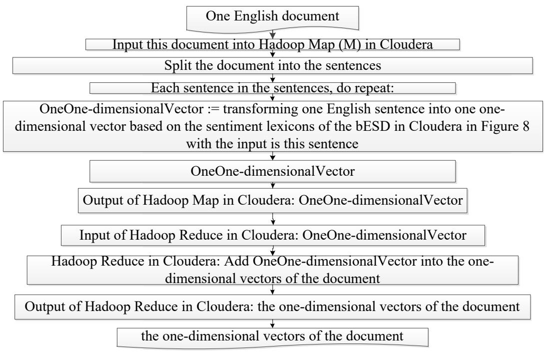

FIGURE 9 Overview of transferring all the sentences of one document of the testing data set into the one-dimensional vectors of the document of testing data set based on the sentiment lexicons of the bESD in the parallel network environment

We build the algorithm 13 to perform the Hadoop Map phase of transferring all the sentences of one document of the testing data set into the one-dimensional vectors of the document of testing data set based on the sentiment lexicons of the bESD in the Cloudera parallel network environment . The main ideas of the algorithm 13 are as follows:

Input: one document of the testing data set;

Output: one one-dimensional vector of this document

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

Step 2: Split this document into the sentences;

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

Step 4: one one-dimensional vector := The transforming one English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera Figure 7 with the input is this sentence

Step 5: Return one one-dimensional vector; //the output of the Hadoop Map phase.

We propose the algorithm 14 to perform the Hadoop Reduce phase of transferring all the sentences of one document of the testing data set into the one-dimensional vectors of the document of testing data set based on the sentiment lexicons of the bESD in the Cloudera parallel network environment. The main ideas of the algorithm 14 are as follows:

Input: one one-dimensional vector of this document

Output: the one-dimensional vectors of this document

Step 1: Receive one one-dimensional vector;

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

Step 3: Return the one-dimensional vectors of this document;

Figure 10, we transfer the positive sentences of the training data set into the positive one-dimensional vectors (called the positive vector group of the training data set) in the distributed system.

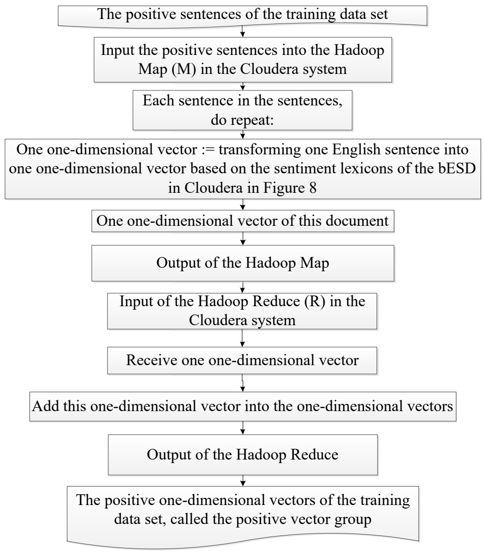

FIGURE 10 Overview of transferring the positive sentences of the training data set into the positive one-dimensional vectors (called the positive vector group of the training data set) in the distributed system.

Figure 10, the stage 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 positive sentences of the training data set. The output of the Hadoop Map phase is one one-dimensional vector of the positive sentences of the training data set. The input of the Hadoop Reduce phase is the output of the Hadoop Map phase, thus, the input of the Hadoop Reduce phase is one one-dimensional vector of one sentence of the positive sentences of the training data set). The output of the Hadoop Reduce phase is the positive one-dimensional vectors, called the positive vector group (corresponding to the positive sentences of the training data set)

We propose the algorithm 15 to perform the Hadoop Map phase of transferring the positive sentences of the training data set into the positive one-dimensional vectors (called the positive vector group of the training data set) in the distributed system Figure 10. The main ideas of the algorithm 15 are as follows:

Input: the positive sentences of the training data set

Output: one one-dimensional vector of the positive sentences of the training data set

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

Step 2: Each sentences in the positive sentences, do repeat:

Step 3: OneOne-DimensionalVector := The transforming one English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera Figure 7.

Step 4: Return OneOne-DimensionalVector;

We propose the algorithm 16 to implement the Hadoop Reduce phase of transferring the positive sentences of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) in the distributed system Figure 10. The main ideas of the algorithm 16 are as follows:

Input: one one-dimensional vector of the positive sentences of the training data set.

Output: the positive one-dimensional vectors, called the positive vector group (corresponding to the positive sentences of the training data set).

Step 1: Receive one one-dimensional vector;

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

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

Figure 11, we transfer the negative sentences of the training data set into the negative one-dimensional vectors (called the negative vector group of the training data set) in the distributed system.

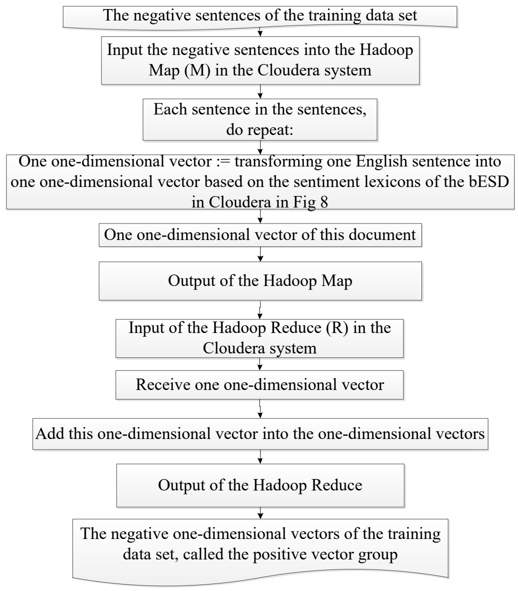

FIGURE 11 Overview of transferring the negative sentences of the training data set into the negative one-dimensional vectors (called the negative vector group of the training data set) in the distributed system.

Figure 11, the stage 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 negative sentences of the training data set. The output of the Hadoop Map phase is one one-dimensional vector of the negative sentences of the training data set. The input of the Hadoop Redude phase is the output of the Hadoop Map phase, thus, the input of the Hadoop Reduce phase is one one-dimensional vector of one sentence of the negative sentences of the training data set). The output of the Hadoop Reduce phase is the positive one-dimensional vectors, called the negative vector group (corresponding to the negative sentences of the training data set)

We propose the algorithm 17 to perform the Hadoop Map phase of transferring the negative sentences of the training data set into the negative one-dimensionalvectors (called the negative vector group of the training data set) in the distributed system Figure 11. The main ideas of the algorithm 17 are as follows:

Input: the negative sentences of the training data set

Output: one one-dimensionalvector of the negative sentences of the training data set

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

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

Step 3: OneOne-DimensionalVector := the transforming one English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera Figure 7

Step 4: Return OneOne-DimensionalVector;

We propose the algorithm 18 to implement the Hadoop Reduce phase of transferring the negative sentences of the training data set into the negative multi-dimensional vectors (called the negative vector group of the training data set) in the distributed system Figure 11. The main ideas of the algorithm 18 are as follows:

Input: one one-dimensional vector of the negative sentences of the training data set

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

Step 1: Receive one one-dimensional vector;

Step 2: Add this one-dimensional vector into NegativeVectorGroup;

Step 3: Return NegativeVectorGroup - the negative one-dimensional vectors, called the negative vector group (corresponding to the negative sentences of the training data set);

Figure 12, we use the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the parallel environment.

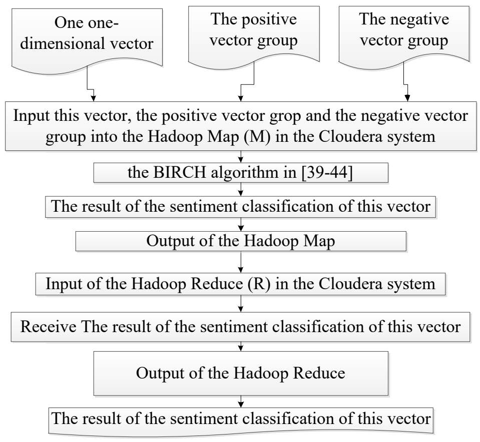

FIGURE 12 Overview of using the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the parallel environment

Figure 12, this stage has two phases as follows: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map is one one-dimensional vector (corresponding one sentence of one document of the testing data set), the positive vector grop and the negative vector group the training data set. The output of the Hadoop Map is the result of the sentiment classification of this vector. The input of the Hadoop Reduce is the output of the Hadoop Map, thus the input of the Hadoop Reduce is the result of the sentiment classification of this vector. The output of the Hadoop Reduce is the result of the sentiment classification of this vector.

We propose the algorithm 19 to perform the Hadoop Map phase of using the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the parallel environment. The main ideas of the algorithm 19 are as follows:

Input: one one-dimensional vector of a document in the testing data set; the positive vector group and the negative vector group of the training data set.

Output: the result of clustering the vector into either the positive vector group or the negative vector group.

Step 1: Scan all data and build an initial in-memory CF tree, using the given amount of memory and recycling space on disk.

Step 2: With each vector in n vectors, do:

Step 3: Condense into desirable length by building a smaller CF tree.

Step 4: Global clustering with the vector into CF Triple 1 or CF Triple 2

Step 5: Cluster refining – this is optional, and requires more passes over the data to refine the results.

Step 6: Return the result of clustering the vector into either the positive vector group or the negative vector group;// the output of the Hadoop Map

We build the algorithm 20 to implement the Hadoop Reduce phase of using the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the parallel environment. The main ideas of the algorithm 20 are as follows:

Input: the result of clustering the vector into either the positive vector group or the negative vector group – the output of the Hadoop Map

Output: the result of clustering the vector into either the positive vector group or the negative vector group.

Step 1: Receive the result of clustering the vector into either the positive vector group or the negative vector group;

Step 2: Return the result of clustering the vector into either the positive vector group or the negative vector group;

Figure 13, we use the BIRCH and the one-dimensional vectors to cluster one document of the testing data set into either the positive or the negative in the distributed environment. The input of the Hadoop Map is one document of the testing data set, the positive vector group and the negative vector group of the training data set. The output of the Hadoop Map is the result of the sentiment classification of one one-dimensional vector (corresponding to one sentence of this document) into either the positive vector group or the negative vector group. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is the result of the sentiment classification of one one-dimensional vector (corresponding to one sentence of this document) into either the positive vector group or the negative vector group. The output of the Hadoop Reduce is the result of the sentiment classification of this document.

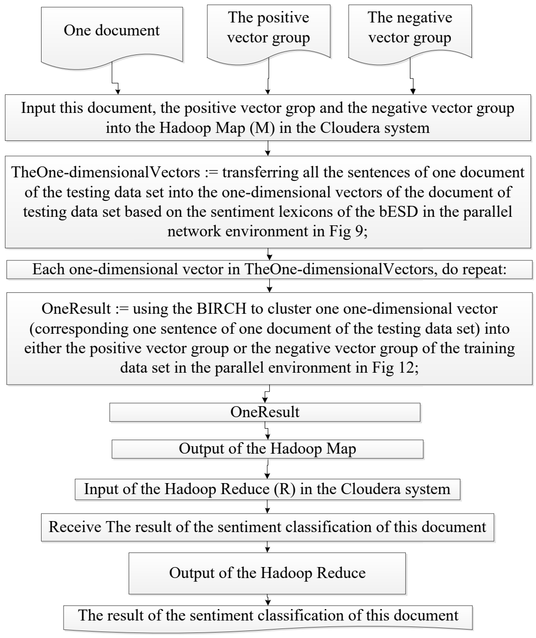

FIGURE 13 Overview of using the BIRCH and the one-dimensional vectors to cluster one document of the testing data set into either the positive or the negative in the distributed environment

We propose the algorithm 21 to perform the Hadoop Map phase of use the BIRCH and the one-dimensional vectors to cluster one document of the testing data set into either the positive or the negative in the distributed environment. The main ideas of the algorithm 21 are as follows:

Input: one document of the testing data set; the positive vector group and the negative vector group of the training data set.

Output: the result of the sentiment classification of one one-dimensional vector (corresponding to one sentence of this document) into either the positive vector group or the negative vector group:

Step 1: Input this document, the positive vector group and the negative vector group into the Hadoop Map in the Cloudera system.

Step 2: TheOne-dimensionalVectors := transferring all the sentences of one document of the testing data set into the one-dimensional vectors of the document of testing data set based on the sentiment lexicons of the bESD in the parallel network environment Figure 9;

Step 3: Each one-dimensional vector in TheOne-dimensionalVectors, do repeat:

Step 4: OneResult := using the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the parallel environment Figure 12;

Step 5: Return OneResult; // the output of the Hadoop Map

We propose the algorithm 22 to perform the Hadoop Reduce phase of use the BIRCH and the one-dimensional vectors to cluster one document of the testing data set into either the positive or the negative in the distributed environment. The main ideas of the algorithm 22 are as follows:

Input: OneResult - the result of the sentiment classification of one one-dimensional vector (corresponding to one sentence of this document) into either the positive vector group or the negative vector group

Output: the result of the sentiment classification of this document.

Step 1: Receive OneResult - the result of the sentiment classification of one one-dimensional vector (corresponding to one sentence of this document) into either the positive vector group or the negative vector group;

Step 2: Add OneResult into the result of the sentiment classification of this document;

Step 3: Return the result of the sentiment classification of this document;

Figure 14, we use the BIRCH and the one-dimensional vectors to cluster the documents of the testing data set into either the positive or the negative in the parallel network environment. This stage comprises two phases: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map is the documents of the testing data set and the training data set. The output of the Hadoop Map is the result of the sentiment classification of one document of the testing data set. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is the result of the sentiment classification of one document of the testing data set. The output of the Hadoop Reduce is the results of the sentiment classification of the documents of the testing data set.

We build the algorithm 23 to implement the Hadoop Map phase of clustering the documents of the testing data set into either the positive or the negative in the sequential environment. The main ideas of the algorithm 23 are as follows:

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

Output: the result of the sentiment classification of one document of the testing data set;

Step 1: transferring the positive sentences of the training data set into the positive one-dimensional vectors (called the positive vector group of the training data set) in the distributed system Figure 10

Step 2: transferring the negative sentences of the training data set into the negative one-dimensional vectors (called the negative vector group of the training data set) in the distributed system Figure 101

Step 3: Input the documents of the testing data set, the positive vector group and the negative vector group into the Hadoop Map in the Cloudera system

Step 4: Each document in the documents of the testing data set, do repeat:

Step 5: OneResult := using the BIRCH and the one-dimensional vectors to cluster one document of the testing data set into either the positive or the negative in the distributed environment Figure 13 with the input is this document, the positive vector group and the negative vector group.

Step 6: Return OneResult - the result of the sentiment classification of one document of the testing data set;//the output of the Hadoop Map

We build the algorithm 24 to perform the Hadoop Reduce phase of clustering the documents of the testing data set into either the positive or the negative in the parallel environment. The main ideas of the algorithm 24 are as follows:

Input: OneResult - the result of the sentiment classification of one document of the testing data set;//the output of the Hadoop Map

Output: the results of the sentiment classification of the documents of the testing data set;

Step 1: Receive OneResult ;

Step 2: Add OneResult into the results of the sentiment classification of the documents of the testing data set;

Step 3: Return the results of the sentiment classification of the documents of the testing data set;

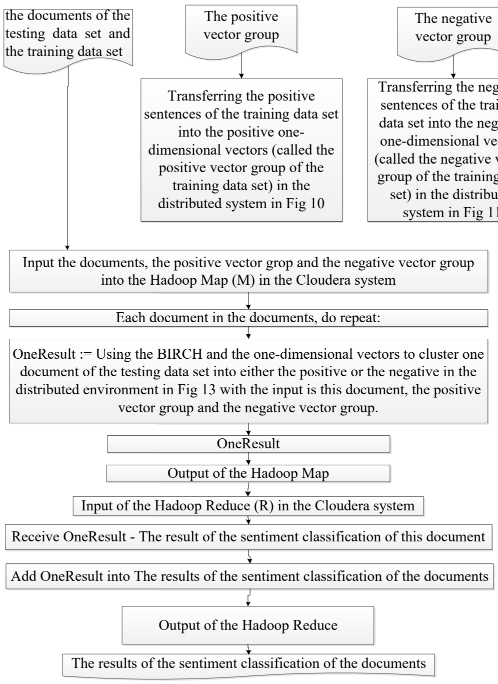

FIGURE 14 Overview of using the BIRCH and the one-dimensional vectors to cluster the documents of the testing data set into either the positive or the negative in the parallel network environment.

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 6,500,000,000 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 our model related to the BIRCH and the one-dimensional vectors. The configuration of the server in the sequential environment is: Inel® Server Board S1200V3RPS, Inel® Pentium® Processor G3220 (3M Cache, 3.00 GHz), 2GB BUBC3-10600 EBUBC 1333 MHz LP Unbuffered DIMMs. The operating system of the server is: Cloudera. We perform the proposed model related to the BIRCH and the one-dimensional vectors 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 proposed model related to the BIRCH and the one-dimensional vectors in the Cloudera. The configuration of each server in the Cloudera system is: Inel® Server Board S1200V3RPS, Inel® Pentium® Processor G3220 (3M Cache, 3.00 GHz), 2GB BUBC3-10600 EBUBC 1333 MHz LP Unbuffered DIMMs. The operating system of each server in the 9 servers is: Cloudera. All 9 nodes have the same configuration information.

The results of the documents of the English testing data set to test are presented in Table 5 below.

The accuracy of the emotional classification of the documents in the English testing data set is shown in Table 6 below.

In Table 7 below, the average time of the classification of our new model for the English documents in testing data set are displayed

6 Conclusion

Although our new model has been tested on our English data set, it can be applied to many other languages. In this paper, our model has been tested on the 6,500,000,000 English documents of the testing data set in 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 new model to classify sentiment of English documents using the BIRCH and the one-dimensional vectors with Hadoop Map (M) /Reduce (R) in the Cloudera parallel network environment. With our proposed new model, we have achieved 87.76% accuracy of the testing data set in Table 6. 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 7, the average time of the semantic classification of using the BIRCH and the one-dimensional vectors in the sequential environment is 30,302,469,000 seconds / 6,500,000,000 English documents and it is greater than the average time of the emotion classification of using the BIRCH and the one-dimensional vectors in the Cloudera parallel network environment – 3 nodes which is 9,100,823,000 seconds / 6,500,000,000 English documents. The average time of the emotion classification of using the BIRCH and the one-dimensional vectors in the Cloudera parallel network environment – 9 nodes, which is 3,500,274,000 seconds / 6,500,000,000 English documents, is the shortest time. Besides, the average time of the emotion classification of using the BIRCH and the one-dimensional vectors in the Cloudera parallel network environment – 6 nodes is 5,350,411,000 seconds / 6,500,000,000 English documents

The execution time of using the BIRCH and the one-dimensional vectors 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 using the BIRCH and the one-dimensional vectors to classify semantics of English documents based on sentences. The proposed model can process millions of documents in the shortest time. This study can be performed in distributed systems to shorten the execution time of the proposed model. It can be applied to other languages. Its negatives are as follows: It has a low rate of accuracy. It costs too much and takes too much time to implement this proposed model.

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

In Table 8, the comparisons of our model's results with the works in [39-44] are presented.

The comparisons of our model's advantages and disadvantages with the works in [39-44] are displayed in Table 9.

In Table 10, the comparisons of our model's results with the works in [50, 51, 52] are shown.

The comparisons of our model's advantages and disadvantages with the works in [50, 51, 52] are presented in Table 11.

In Table 12, the comparisons of our model with the latest sentiment classification models (or the latest sentiment classification methods) in [53-63] are displayed.

The comparisons of our model's positives and negatives with the latest sentiment classification models (or the latest sentiment classification methods) in [53-63] are shown in Table 13.

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 analyse the emotions of texts and speeches, and machines that can analyse sentiments.

References

[1] Bai A, Hammer H 2014 Constructing sentiment lexicons in Norwegian from a large text corpus *2014 IEEE 17th Inernational Conference on Computational Science and Engineering*

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

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

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

[5] Jovanoski D, Pachovski V, Nakov P 2015 Sentiment Analysis in Twitter for Macedonian *Proceedings of Recent Advances in Natural Language Processing, Bulgaria* 249–57

[6] Htait A, Fournier S, Bellot P 2016 LSIS at SemEval-2016 Task 7: Using Web Search Engines for English and Arabic Unsupervised Sentiment Intensity Prediction *Proceedings of SemEval-2016, California* 481–5

[7] Wan Xiaojun 2009 Co-Training for Cross-Lingual Sentiment Classification *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, Singapore* 235–43

[8] Brooke J, Tofiloski M, Taboada M 2009 Cross-Linguistic Sentiment Analysis: From English to Spanish *International Conference RANLP 2009 - Borovets, Bulgaria* 50–4

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

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

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

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

[13] Guangwei Wang, Kenji Araki 2007 SO-PMI for Japanese Weblog Opinion Mining by Using a Balancing Factor and Detecting Neutral Expressions *Proceedings of NAACL HLT 2007, Companion Volume, NY* 189–92

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

[15] Nguyen Thi Thu An, Masafumi Hagiwara 2014 Adjective-Based Estimation of Short Sentence's Impression. (KEER2014) *Proceedings of the 5th Kanesi Engineering and Emotion Research; International*

Conference; Sweden

[16] Nihalahmad R Shikalgar, Arati M Dixit 2014 JIBCA: Jaccard Index based Clustering Algorithm for Mining Online Review *International Journal of Computer Applications (0975 – 8887)* **105**(15)

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

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

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

[20] Ren Y, Kaji N, Yoshinaga N, Kitsuregaw M 2014 Sentiment Classification in Under-Resourced Languages Using Graph-based Semi-supervised Learning Methods *IEICE Trans. Inf. & Syst.* **E97–D**(4) DOI: 10.1587/transinf.E97.D.1

[21] Oded Netzer, Ronen Feldman, Jacob Goldenberg, Moshe Fresko 2012 Mine Your Own Business: Market-Structure Surveillance Through Text Mining *Marketing Science* **31**(3) 521-43

[22] Yong Ren, Nobuhiro Kaji, Naoki Yoshinaga, Masashi Toyoda, Masaru Kitsuregawa 2011 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* 420 - 9

[23] José Alfredo Hernández-Ugalde, Jorge Mora-Urpí, Oscar J. Rocha 2011 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* **58**(4) 571-83

[24] Ponomarenko J V, Bourne P E, Shindyalov I N 2002 Building an automated classification of DNA-binding protein domain *Bioinformatics* **18** S192-S201

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

[26] Mladenović Drinić S, Nikolić Ana, Perić V 2008 Cluster Analysis of Soybean Genotypes Based on RAPD Markers *Proceedings. 43rd Croatian and 3rd International Symposium on Agriculture. Opatija. Croatia* 367-70

[27] Tamás J, Podani J, Csontos P 2001 An extension of presence/absence coefficients to abundance data: a new look at absence *Journal of Vegetation Science* **12** 401-10

[28] Vo Ngoc Phu, Vo Thi Ngoc Chau, Vo Thi Ngoc Tran, Nguyen Duy Dat 2017 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

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

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

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

[32] Vo Ngoc Phu, Vo Thi Ngoc Chau, Vo Thi Ngoc Tran, Nguyen Duy Dat, Khanh Ly Doan Duy 2017 Semantic Lexicons of English Nouns for Classification *International Journal of Evolving Systems* DOI: 10.1007/s12530-017-9188-6, 69 pages

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

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

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

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

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

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

[39] Tian Zhang, Raghu Ramakrishnan, Miron Livny 1996 BIRCH: an efficient data clustering method for very large databases. SIGMOD '96 *Proceedings of the 1996 ACM SIGMOD international conference on Management of data, Pages 103-114, Montreal, Quebec, Canada*

[40] Tian Zhang, Raghu Ramakrishnan, Miron Livny 1997 BIRCH: A New Data Clustering Algorithm and Its Applications *Data Mining and Knowledge Discovery* **1**(2) 141–82

[41] Jörg Sander, Martin Ester, Hans-Peter Kriegel, Xiaowei Xu 1998 Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications *Data Mining and Knowledge Discovery* **2**(2) 169–94

[42] Vijaya P A, Narasimha Murty M, Subramanian D K 2004 Leaders–Subleaders: An efficient hierarchical clustering algorithm for large data sets *Pattern Recognition Letters* https://doi.org/10.1016/j.patrec.2003.12.013 **25**(4) 505-13

[43] Jörg Sander, Martin Ester, Hans-Peter Kriegel, Xiaowei Xu 1998 Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications *Data Mining and Knowledge Discovery* **2**(2) 169–94

[44] Ackermann M R, Märtens M, Raupach C, Swierkot K, Lammersen C, Sohler C 2012 StreamKM++: A clustering algorithm for data streams *Journal of Experimental Algorithmics (JEA)* **1**, Article No. 2.4, ACM New York, NY, USA

[45] Seung-Seok Choi, Sung-Hyuk Cha, Charles C. Tappert 2010 A Survey of Binary Similarity and Distance Measures *Systemics, Cybernetics and Informatics* ISSN: 1690-4524 **8**(1)

[46] Ulfendahl M, Chan E, BARONI-URBANI & BUSER-II W B, Prost-Domasky Scott, Elson E L 1998 Axial and transverse stiffness measures of cochlear outer hair cells suggest a common mechanical basis *Pflügers*

Archiv **436**(1) 9–15

[47] BARONI-URBANI & BUSER-II W B, Petersen N O 2008 Cell poker: An apparatus for stress-strain measurements on living cells *Review of Scientific Instruments* **51**(5) 10.1063/1.1136256

[48] BARONI-URBANI & BUSER-II P K, Duxbury J M 1985 Introduction of Acetylene into Soil for Measurement of Denitrification Soil *Science Society of America Journal* **50**(1) 260-3

[49] Zdenek Hubálek 1982 Coefficients Of Association And Similarity, Based On Binary (Presence-Absence) Data: An Evaluation *Biological Reviews* **57**(4) 669–89 DOI: 10.1111/j.1469-185X.1982.tb00376.x

[50] Sang-Tae Kim, O'Neil J R 1997 Equilibrium and nonequilibrium oxygen isotope effects in synthetic carbonates *Geochimica et Cosmochimica Acta* **61**(16) 3461-75 https://doi.org/10.1016/S0016-7037(97)00169-5

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

[52] Carrera-Trejo V, Sidorov G, Miranda-Jiménez S, Moreno Ibarra M, Cadena Martínez R 2015 Latent Dirichlet Allocation complement in the vector space model for Multi-Label Text Classification *International Journal of Combinatorial Optimization Problems and Informatics* **6**(1) 7-19

[53] Pascal Soucy, Guy W. Mineau 2015 Beyond TFIDF Weighting for Text Categorization in the Vector Space Model *Proceedings of the 19th International Join Conference on Artificial Intelligence, 1130-5, USA*

[54] Basant Agarwal, Namita Mittal 2016 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

[55] Basant Agarwal, Namita Mittal 2016 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

[56] Sérgio Canuto, Marcos André, Gonçalves, Fabrício Benevenuto 2016 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*

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

[58] Vo Ngoc Phu, Phan Thi Tuoi 2014 Sentiment classification using Enhanced Contextual Valence Shifters *International Conference on Asian Language Processing (IALP)* 224-9

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

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

[61] Vo Ngoc Phu, Nguyen Duy Dat, Vo Thi Ngoc Tran, Vo Thi Ngoc Tran 2016 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

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

[63] Vo Ngoc Phu, Vo Thi Ngoc Chau, Vo Thi Ngoc Tran 2017 SVM for English Semantic Classification in Parallel Environment *International Journal of Speech Technology (IJST)* 10.1007/s10772-017-9421-5, 31 pages, May 2017

[64] Vo Ngoc Phu, Vo Thi Ngoc Tran, Vo Thi Ngoc Chau, Nguyen Duy Dat, Khanh Ly Doan Duy 2017 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].

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

Table 3: Comparisons of our model's results with the works related to the BARONI-URBANI & BUSER-II coefficient (BUBC) in [45-50]

Table 4: Comparisons of our model's benefits and drawbacks with the works related to the BARONI-URBANI & BUSER-II coefficient (BUBC) in [45-50]

Table 5: The results of the English documents in the testing data set.

Table 6: The accuracy of our new model for the English documents in the testing data set.

Table 7: Average time of the classification of our new model for the English documents in testing data set.

Table 8: Comparisons of our model's results with the works in [39-44]

Table 9: Comparisons of our model's advantages and disadvantages with the works in [39-44]

Table 10: Comparisons of our model's results with the works in [51-53]

Table 11: Comparisons of our model's advantages and disadvantages with the works in [51-53]

Table 12: Comparisons of our model with the latest sentiment classification models (or the latest sentiment classification methods) in [54-64]

Table 13: Comparisons of our model's positives and negatives with the latest sentiment classification models (or the latest sentiment classification methods) in [54-64]

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

BARONI-URBANI & BUSER-II coefficient (BUBC)
Semantic classification, sentiment classification: SC

Studies	PMI	JM	Language	SD	DT	BUBC	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 inhe 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); RoBUBChio; 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

Studies	PMI	JM	Language	SD	DT	BUBC	SC	Other measures	Search engines
[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 BUBCKeown (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.

Surveys	Approach	Advantages	Disadvantages
[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 analysing/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 unlabelled 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 analysing 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 analysing 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 neighbour, 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

Surveys	Approach	Advantages	Disadvantages
[11]	Adapting Information Bottleneck Method for Automatic Construction of Domain-oriented Sentiment Lexicon	The authors' theory verifies the convergence property of the proposed method. The empirical results also support the authors' theoretical analysis. In their experiment, it is shown that proposed method greatly outperforms the baseline methods in the task of building out-of-domain sentiment lexicon.	In this study, only the mutual information measure is employed to measure the three kinds of relationship. In order to show the robustness of the framework, the authors' future effort is to investigate how to integrate more measures into this framework.
[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 analyse 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 stimation of Short Sentence's Impression	The adjectives are ranked and top na adjectives are considered as an output of system. For example, the experiments were carried out and got fairly good results. With the input "it is snowy", the results are white (0.70), light (0.49), cold (0.43), solid (0.38), and scenic (0.37)	In the authors' future work, they will improve more in the tasks of keyword extraction and semantic similarity methods to make the proposed system working well with complex inputs.
[16]	Jaccard Index based Clustering Algorithm for Mining Online Review	In this work, the problem of predicting sales performance using sentiment information mined from reviews is studied and a novel JIBCA Algorithm is proposed and mathematically modelled. The outcome of this generates knowledge from mined data that can be useful for forecasting sales.	For future work, by using this framework, it can extend it to predicting sales performance in the other domains like customer electronics, mobile phones, computers based on the user reviews posted on the websites, etc.
[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

Surveys	Approach	Advantages	Disadvantages
[18]	Ensemble of Classification Algorithm 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 labelled vertices in aBUBCordance 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 unlabelled data to fully take advantage of SSL Algorithm
[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 resource scarce setting, the stable performance of parameter-free label propagation is promising.	The authors plan to further improve the performance of LP in sentiment classification, especially when the authors only have a small number of labelled seeds. The authors will exploit the idea of restricting the label propagating steps when the available labelled data is quite small.
[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.

Surveys	Approach	Advantages	Disadvantages
[29]	A Valences-Totalling 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-Totalling Model for Vietnamese Sentiment Classification	The authors have used the VTMfV to classify 30,000 Vietnamese documents, which include the 15,000 positive Vietnamese documents and the 15,000 negative Vietnamese documents. The authors have achieved accuracy in 63.9% of the authors' Vietnamese testing data set. VTMfV is not dependent on the special domain. VTMfV is also not dependent on the training data set and there is no training stage in this VTMfV. From the authors' results in this work, our VTMfV can be applied in the different fields of the Vietnamese natural language processing. In addition, the authors' TCMfV can be applied to many other languages such as Spanish, Korean, etc. It can also be applied to the big data set sentiment classification in Vietnamese and can classify millions of the Vietnamese documents	it has a low accuracy.
[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 adverbs, etc.
Our work	-We use the BIRCH and the one-dimensional vectors to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. -The advantages and disadvantages of this survey are shown in the Conclusion section.		

TABLE 3 Comparisons of our model's results with the works related to the BARONI-URBANI & BUSER-II coefficient (BUBC) in [45-50]

Studies	PMI	JM	BARONI-URBANI & BUSER-II coefficient (BUBC)	Language	SD	DT	Sentiment Classification
[45]	Yes	Yes	Yes	English	NM	NM	No mention
[46]	No	No	Yes	NM	NM	NM	No mention
[47]	No	No	Yes	NM	NM	NM	No mention
[48]	No	No	Yes	NM	NM	NM	No mention
[49]	No	No	Yes	NM	NM	NM	No mention
[50]	No	No	Yes	NM	NM	NM	No mention
Our work	No	No	Yes	English Language	No	No	Yes

TABLE 4 Comparisons of our model's benefits and drawbacks with the studies related to the BARONI-URBANI & BUSER-II coefficient (BUBC) in [45-50]

Surveys	Approach	Benefits	Drawbacks
[45]	A Survey of Binary Similarity and Distance Measures	Applying appropriate measures results in more accurate data analysis. Notwithstanding, few comprehensive surveys on binary measures have been conducted. Hence the authors collected 76 binary similarity and distance measures used over the last century and reveal their correlations through the hierarchical clustering technique	No mention
[46]	Axial and transverse stiffness measures of cochlear outer hair cells suggest a common mechanical basis	The results imply that the two different stiffness properties may originate from the same cytoskeletal structures. It is suggested that the mechanical properties of the outer hair cells are designed to influence the sound-induced motion of the reticular lamina. In such a system, stiffness changes of the outer hair cell bodies could actively control the efficiency of the mechanical coupling between the basilar membrane and the important mechanoelectrical transduction sites at the surface of the hearing organ.	No mention
[47]	Cell poker: An apparatus for stress-strain measurements on living cells	The resulting deformation is deduced from a solution conductance measurement with a sensitivity of 0.1 μm. The technique is compatible with optical microscopy and electrophysiological recording	No mention
[48]	Introduction of Acetylene Into Soil for Measurement of Denitrification	This survey describes a method for the introduction of C_2H_2 at low concentrations into the soil by diffusion. The method utilizes an above-ground manifold connected to diffusion tubes inserted in the soil and C_2H_2 flows through the manifold at a rate of 0.33 mL s^{-1}. Acetylene levels in the soil atmosphere within the flux chamber site ranged from 0.16 to 2.07 mol m^{-3}, and the apparatus did not alter CO_2 fluxes as measured by a static chamber method.	No mention
[49]	Coefficients of Association and Similarity, Based on Binary (Presence-Absence) Data: an Evaluation	For some purposes, however, other 'admissible' coefficients would be more optimal, and the choice of a measure should be related to the nature of the data. It is tentatively suggested that three or so alternative coefficients be used and the results compared on the same data basis; moreover, significance tests on association should be carried out whenever possible.	No mention
[50]	Equilibrium and none quilibrium oxygen isotope effects in synthetic carbonates	The ability to produce nonequilibrium carbonates allowed assessment to be made, for the first time, of the temperature dependence of none quilibrium stable isotope fractionations in mineral systems. The temperature coefficients of a(carbonate-water) for none quilibrium divalent metal carbonates are greater than those for equilibrium carbonates, a finding that may bear on the interpretation of analyses of biogenic carbonates forming out of isotopic equilibrium in nature	No mention
Our work	-We use the BIRCH and the one-dimensional vectors to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. -The advantages and disadvantages of this survey are shown in the Conclusion section.		

TABLE 5 The results of the English documents in the testing data set

	Testing Dataset	Correct Classification	Incorrect Classification
Negative	3,250,000,000	2,862,186,000	387,814,000
Positive	3,250,000,000	2,842,214,000	407,786,000
Summary	6,500,000,000	5,704,400,000	795,600,000

TABLE 6 The accuracy of our novel model for the English documents in the testing data set

Proposed Model	Class	Accuracy
Our new model	Negative Positive	87.76 %

TABLE 7 Average time of the classification of our novel model for the documents in testing data set

	Average time of the classification /6,500,000,000 documents
The novel model in the sequential environment	30,302,469,000 seconds
The novel model in the Cloudera distributed system with 3 nodes	9,100,823,000 seconds
The novel model in the Cloudera distributed system with 6 nodes	5,350,411,000 seconds
The novel model in the Cloudera distributed system with 9 nodes	3,500,274,000 seconds

TABLE 8 Comparisons of our model's results with the works in [39-44]

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

Studies	BUBC	CT	Sentiment Classification	PNS	SD	DT	Language	VSM
[39]	No	No	No	No	Yes	No	EL	Yes
[40]	No	No	Yes	No	Yes	No	EL	Yes
[41]	No	No	Yes	No	Yes	Yes	EL	Yes
[42]								
[43]								
[44]								
Our work	Yes	Yes	Yes	Yes	No	No	EL	Yes

TABLE 9 Comparisons of our model's advantages and disadvantages with the works in [39-44]

Researches	Approach	Advantages	Disadvantages
[39]	BIRCH: an efficient data clustering method for very large databases	This survey presents a data clustering method named BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), and demonstrates that it is especially suitable for very large databases. BIRCH incrementally and dynamically clusters incoming multi-dimensional metric data points to try to produce the best quality clustering with the available resources (i.e., available memory and time constrains). BIRCH can typically find a good clustering with a single scan of the data, and improve the quality further with a few additional scans. BIRCH is also the first clustering algorithm proposed in the database area to handle "noise" (data points that are not part of the underlying pattern) effectively. The authors evaluate BIRCH's time/space efficiency, data input order sensitivity, and clustering quality through several experiments. The authors also present a performance comparisons of BIRCH versus CLARANS, a clustering method proposed recently for large datasets, and show that BIRCH is consistently superior.	No mention
[40]	BIRCH: A New Data Clustering Algorithm and Its Applications	In this survey, an efficient and scalable data clustering method is proposed, based on a new in-memory data structure called CF-tree, which serves as an in-memory summary of the data distribution. The authors have implemented it in a system called BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), and studied its performance extensively in terms of memory requirements, running time, clustering quality, stability and scalability; the authors also compare it with other available methods. Finally, BIRCH is applied to solve two real-life problems: one is building an iterative and ineractive pixel classification tool, and the other is generating the initial codebook for image compression.	No mention
[41]	Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications	In this survey, the authors generalize this algorithm in two important directions. The generalized algorithm—called GDBSCAN—can cluster poin objects as well as spatially extended objects according to both, their spatial and their nonspatial attributes. In addition, four applications using 2D poins (astronomy), 3D poins (biology), 5D poins (earth science) and 2D polygons (geography) are presented, demonstrating the applicability of GDBSCAN to real-world problems.	No mention
[42]	Leaders–Subleaders: An efficient hierarchical clustering algorithm for large data sets	As an example, a two level clustering algorithm—'Leaders–Subleaders', an extension of the leader algorithm is presented. Classification accuracy (CA) obtained using the representatives generated by the Leaders–Subleaders method is found to be better than that of using leaders as representatives. Even if more number of prototypes are generated, classification time is less as only a part of the hierarchical structure is searched.	No mention
[43]	Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications	In this survey, the authors generalize this algorithm in two important directions. The generalized algorithm—called GDBSCAN—can cluster point objects as well as spatially extended objects according to both, their spatial and their nonspatial attributes. In addition, four applications using 2D poins (astronomy), 3D poins (biology), 5D poins (earth science) and 2D polygons (geography) are presented, demonstrating the applicability of GDBSCAN to real-world problems.	No mention

Researches	Approach	Advantages	Disadvantages
[44]	StreamKM++: A clustering algorithm for data streams	The authors compare the authors' algorithm experimentally with two well-known streaming implementations: BIRCH and StreamLS. In terms of quality (sum of squared errors), the authors' algorithm is comparable with StreamLS and significantly better than BIRCH (up to a factor of 2). Besides, BIRCH requires significant effort to tune its parameters. In terms of running time, the authors' algorithm is slower than BIRCH. Comparing the running time with StreamLS, it turns out that the authors' algorithm scalesmuch better with increasing number of centers. The authors conclude that, if the first priority is the quality of the clustering, then the authors' algorithm provides a good alternative to BIRCH and StreamLS, in particular, if the number of cluster centers is large. The authors also give a theoretical justification of the authors' approach by proving that the authors' sample set is a small coreset in low-dimensional spaces.	No mention
Our work	-We use the BIRCH and the one-dimensional vectors to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. -The advantages and disadvantages of the proposed model are shown in the Conclusion section.		

TABLE 10 Comparisons of our model's results with the works in [51-53]

Studies	BUBC	CT	Sentiment Classification	PNS	SD	DT	Language	VSM
[51]	No	No	No	No	Yes	No	EL	Yes
[52]	No	No	Yes	No	Yes	No	EL	Yes
[53]	No	No	Yes	No	Yes	Yes	EL	Yes
Our work	Yes	Yes	Yes	Yes	No	No	EL	Yes

TABLE 11 Comparisons of our model's advantages and disadvantages with the works in [51-53]

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

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

TABLE 12 Comparisons of our model with the latest sentiment classification models (or the latest sentiment classification methods) in [54-64]

Studies	BUBC	CT	Sentiment Classification	PNS	SD	DT	Language	VSM
[54]	No	No	Yes	NM	Yes	Yes	Yes	vector
[55]	No	No	Yes	NM	Yes	Yes	NM	NM
[56]	No	No	Yes	NM	Yes	Yes	EL	NM
[57]	No	No	Yes	NM	Yes	Yes	NM	NM
[58]	No	No	Yes	No	No	No	EL	No
[59]	No	No	Yes	No	No	No	EL	No
Our work	Yes	Yes	Yes	Yes	No	No	Yes	Yes

TABLE 13 Comparisons of our model's positives and negatives the latest sentiment classification models (or the latest sentiment classification methods) in [54-64]

Studies	Approach	Positives	Negatives
[54]	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
[55]	Semantic Orientation-Based Approach for Sentiment Analysis	This approach initially mines sentiment-bearing terms from the unstructured text and further computes the polarity of the terms. Most of the sentiment-bearing terms are multi-word features unlike bag-of-words, e.g., "good movie," "nice cinematography," "nice actors," etc. Performance of semantic orientation-based approach has been limited in the literature due to inadequate coverage of multi-word features.	No mention

Studies	Approach	Positives	Negatives
[56]	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 Algorithm and feature selection techniques in different sentiment analysis tasks such as scoring movies or products According to their related reviews.
[57]	Rule-Based Machine Learning Algorithm	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 Algorithm have been performed through a ten-fold cross validation training model for sentiment classification.	No mention
[58]	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
[59]	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 BIRCH and the one-dimensional vectors to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. -The positives and negatives of the proposed model are given in the Conclusion section.		

Appendix of Codes

ALGORITHM 1: Creating a basis English sentiment dictionary (bESD) in a sequential environment
ALGORITHM 2: Implementing the Hadoop Map phase of creating a basis English sentiment dictionary (bESD) in a parallel environment
ALGORITHM 3: Implementing the Hadoop Reduce phase of creating a basis English sentiment dictionary (bESD) in a parallel environment
ALGORITHM 4: transferring one sentence into one-dimensional vector based on the sentiment lexicons of the bESD in the sequential environment
ALGORITHM 5: transferring all the sentences of one document into the one-dimensional vectors based on the sentiment lexicons of the bESD in the sequential system

ALGORITHM 6: transferring all the positive sentences of the training data set into the one-dimensional vector based on the sentiment lexicons of the bESD in the sequential system, called the positive vector group of the training data set
ALGORITHM 7: transferring all the negative sentences of the training data set into the one-dimensional vector based on the sentiment lexicons of the bESD in the sequential system, called the negative vector group of the training data set
ALGORITHM 8: using the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the sequential environment
ALGORITHM 11: clustering one document of the testing data set into either the positive or the negative in the sequential system
ALGORITHM 10: clustering the documents of the testing data set into either the positive or the negative in the sequential environment
ALGORITHM 11: performing the Hadoop Map phase of transforming each English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera
ALGORITHM 12: performing the Hadoop Reduce phase of transforming each English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera
ALGORITHM 15: performing the Hadoop Map phase of transferring all the sentences of one document of the testing data set into the one-dimensional vectors of the document of testing data set based on the sentiment lexicons of the bESD in the Cloudera parallel network environment
ALGORITHM 14: performing the Hadoop Reduce phase of transferring all the sentences of one document of the testing data set into the one-dimensional vectors of the document of testing data set based on the sentiment lexicons of the bESD in the Cloudera parallel network environment
ALGORITHM 15: performing the Hadoop Map phase of transferring the positive sentences of the training data set into the positive one-dimensional vectors (called the positive vector group of the training data set) in the distributed system
ALGORITHM 16: implementing the Hadoop Reduce phase of transferring the positive sentences of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) in the distributed system
ALGORITHM 17: performing the Hadoop Map phase of transferring the negative sentences of the training data set into the negative one-dimensional vectors (called the negative vector group of the training data set) in the distributed system
ALGORITHM 18: implementing the Hadoop Reduce phase of transferring the negative sentences of the training data set into the negative multi-dimensional vectors (called the negative vector group of the training data set) in the distributed system
ALGORITHM 19: performing the Hadoop Map phase of using the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the parallel environment
ALGORITHM 20: implementing the Hadoop Reduce phase of using the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the parallel environment
ALGORITHM 19: performing the Hadoop Map phase of using the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the parallel environment
ALGORITHM 20: implementing the Hadoop Reduce phase of using the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the parallel environment
ALGORITHM 23: implementing the Hadoop Map phase of clustering the documents of the testing data set into either the positive or the negative in the sequential environment
ALGORITHM 24: performing the Hadoop Reduce phase of clustering the documents of the testing data set into either the positive or the negative in the parallel environment

ALGORITHM 1: Creating a basis English sentiment dictionary (bESD) in a sequential environment

Input: the 55,000 English terms; the Google search engine
Output: a basis English sentiment dictionary (bESD)
Begin
Step 1: Set bESD := null;
Step 2: For i = 1; i < 55,000; i++, do repeat:
Step 3: By using eq. (8), eq. (9), and eq. (10) of the calculating a valence of one word (or one phrase) in English in the section (4.1.1), the sentiment score and the polarity of this term i are identified. The valence and the polarity are calculated by using the BUBC through the Google search engine with AND operator and OR operator.
Step 4: Add this term into bESD;
Step 5: End Repeat – End Step 2;
Step 6: Return bESD;
End;

ALGORITHM 2: Implementing the Hadoop Map phase of creating a basis English sentiment dictionary (bESD) in a parallel environment

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

ALGORITHM 3: Implementing the Hadoop Reduce phase of creating a basis English sentiment dictionary (bESD) in a parallel environment

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

ALGORITHM 4: transferring one sentence into one-dimensional vector based on the sentiment lexicons of the bESD in the sequential environment

Input: one English document and the bESD
Output: one one-dimensional vector
Begin
Step 1: Set OneDimensionalVector := {};
Step 2: Set arraySentences := Split the English sentence into many separate terms based on the bESD;
Step 3: For i = 0; i < arraySentences.length; i++, do:
Step 4: Identify the valence of the term arraySentences[i] based on the bESD;
Step 5: Add this term into OneDimensionalVector;
Step 6: If OneDimensionalVector.length is less than m_max Then
Step 7: For j = OneDimensionalVector.length; j< m_max; j++; do:

Step 8: OneDimensionalVector[j] := 0;
Step 9: End For – End Step 7;
Step 10: End If – End Step 6;
Step 11: Return OneDimensionalVector;
End;

ALGORITHM 5: transferring all the sentences of one document into the one-dimensional vectors based on the sentiment lexicons of the bESD in the sequential system

---

Input: one document and the bESD;
Output: the one-dimensional vectors of this document;
Begin
Step 1: Split this document into the sentences;
Step 2: Each sentence in the sentences of this document, do repeat:
Step 3: OneOne-dimensionalVector := The algorithm 4 to transfer one sentence into one-dimensional vector based on the sentiment lexicons of the bESD in the sequential environment with the input is this sentence and the bESD;
Step 4: Add OneOne-dimensionalVector into the one-dimensional vectors of this document;
Step 5: End Repeat – End Step 3;
Step 6: Return the one-dimensional vectors of this document;
End;

ALGORITHM 6: transferring all the positive sentences of the training data set into the one-dimensional vector based on the sentiment lexicons of the bESD in the sequential system, called the positive vector group of the training data set

---

Input: the positive sentences of the training data set and the bESD;
Output: the positive one-dimensional vectors, called the positive vector group of the training data set;
Begin
Step 1: Set the positive vector group := null;
Step 2: Each sentence in the positive sentences of the training data set, do repeat:
Step 3: OneOne-dimensionalVector := The algorithm 4 to transfer one sentence into one-dimensional vector based on the sentiment lexicons of the bESD in the sequential environment with the input is this sentence and the bESD;
Step 4: Add dimensionalVector into the positive vector group;
Step 5: End Repeat – End Step 2;
Step 6: Return the positive vector group;
End;

ALGORITHM 7: transferring all the negative sentences of the training data set into the one-dimensional vector based on the sentiment lexicons of the bESD in the sequential system, called the negative vector group of the training data set

---

Input: the negative sentences of the training data set and the bESD;
Output: the negative one-dimensional vectors, called the negative vector group of the training data set;
Begin
Step 1: Set the negative vector group := null;
Step 2: Each sentence in the negative sentences of the training data set, do repeat:
Step 3: OneOne-dimensionalVector := The algorithm 4 to transfer one sentence into one-dimensional vector based on the sentiment lexicons of the bESD in the sequential environment with the input is this sentence and the bESD;
Step 4: Add dimensionalVector into the negative vector group;
Step 5: End Repeat – End Step 2;
Step 6: Return the negative vector group;

End;

ALGORITHM 8: using the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the sequential environment

---

Input: one one-dimensional vector of a document in the testing data set; the positive vector group and the negative vector group of the training data set.
Output: the result of clustering the vector into either the positive vector group or the negative vector group.
Begin
Step 1: Select randomly k centres (centroid) of k clusters. Each cluster is represented by the centre of this cluster.
Step 2: Calculate the distance between objects to k centres using Euclidean distance.
Step 3: Group objects into the closest group.
Step 4: Identify the new centre of the clusters.
Step 5: Repeat step 2 until no object groups change.
Step 6: Return the result of clustering the vector into either the positive vector group or the negative vector group.
End;

ALGORITHM 9: clustering one document of the testing data set into either the positive or the negative in the sequential system

---

Input: one document of the testing data set; the positive vector group and the negative vector group of the training data set.
Output: The result of the sentiment classification of this document
Begin
Step 1: TheOne-dimensionalVectors := The algorithm 5 to transfer all the sentences of one document into the one-dimensional vectors based on the sentiment lexicons of the bESD in the sequential system with the input is this document;
Step 2: Set count_positive := 0; and count_negative := 0;
Step 3: Each one-dimensional vector in TheOne-dimensionalVectors, do repeat:
Step 4: OneResult := The algorithm 8 to use the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the sequential environment with this vector, the positive vector group and the negative vector group;
Step 5: If OneResults is the positive Then count_positive := count_positive + 1;
Step 6: Else If OneResults is the negative Then count_negative := count_negative + 1;
Step 7: End Repeat – End Step 3;
Step 8: If count_positive is greater than count_negative Then Return positive;
Step 9: Else If count_positive is less than count_negative Then Return negative;
Step 10: Return neutral;
End;

ALGORITHM 10: clustering the documents of the testing data set into either the positive or the negative in the sequential environment

---

Input: the documents of the testing data set and the training data set
Output: the results of the sentiment classification of the documents of the testing data set;
Begin
Step 1: The algorithm 6 to transfer all the positive sentences of the training data set into the one-dimensional vector based on the sentiment lexicons of the bESD in the sequential system, called the positive vector group of the training data set with the input is the positive sentences of the training data set; and the bESD;
Step 2: The algorithm 7 to transfer all the negative sentences of the training data set into the one-

dimensional vector based on the sentiment lexicons of the bESD in the sequential system, called the negative vector group of the training data set with the input is the negative sentences of the training data set; and the bESD;
Step 3: Each document in the documents of the testing data set, do repeat:
Step 4: OneResult := the algorithm 9 to cluster one document of the testing data set into either the positive or the negative in the sequential system with the input is this document, the positive vector group and the negative vector group;
Step 5: Add OneResult into the results of the sentiment classification of the documents of the testing data set;
Step 6: Return the results of the sentiment classification of the documents of the testing data set;
End;

ALGORITHM 11: performing the Hadoop Map phase of transforming each English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera

---

Input: one sentence and the bESD;
Output: one term (one meaningful word/or one meaningful phrase) which the valence is identified
Begin
Step 1: Input this sentence and the bESD into the Hadoop Map in the Cloudera system;
Step 2: Split this sentence into the many meaningful terms (meaningful words/or meaningful phrases) based on the bESD;
Step 3: Each term in the terms, do repeat:
Step 4: Identify the valence of this term based on the bESD;
Step 5: Return this term; //the output of the Hadoop Map phase.
End;

ALGORITHM 12: performingthe Hadoop Reduce phase of transforming each English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera

---

Input: one term (one meaningful word/or one meaningful phrase) which the valence is identified – the output of the Hadoop Map phase
Output: one one-dimensional vector based on the sentiment lexicons of the bESD
Begin
Step 1: Receive one term;
Step 2: Add this term into the one-dimensionalvector;
Step 3: Return the one-dimensionalvector;
End;

ALGORITHM 15: performing the Hadoop Map phase of transferring all the sentences of one document of the testing data set into the one-dimensional vectors of the document of testing data set based on the sentiment lexicons of the bESD in the Cloudera parallel network environment

---

Input: one document of the testing data set;
Output: one one-dimensional vector of this document
Begin
Step 1: Input this document into the Hadoop Map in the Cloudera system;
Step 2: Split this document into the sentences;
Step 3: Each sentence in the sentences, do repeat:
Step 4: one one-dimensional vector := The transforming one English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera Figure 7 with the input is this sentence
Step 5: Return one one-dimensional vector; //the output of the Hadoop Map phase.
End;

ALGORITHM 14: performing the Hadoop Reduce phase of transferring all the sentences of one document of the testing data set into the one-dimensional vectors of the document of testing data

---

set based on the sentiment lexicons of the bESD in the Cloudera parallel network environment

Input: one one-dimensional vector of this document
Output: the one-dimensional vectors of this document
Begin
Step 1: Receive one one-dimensional vector;
Step 2: Add this one-dimensional vector into the one-dimensional vectors of this document;
Step 3: Return the one-dimensional vectors of this document;
End;

ALGORITHM 15: performing the Hadoop Map phase of transferring the positive sentences of the training data set into the positive one-dimensional vectors (called the positive vector group of the training data set) in the distributed system

---

Input: the positive sentences of the training data set
Output: one one-dimensional vector of the positive sentences of the training data set
Begin
Step 1: Input the positive sentences into the Hadoop Map in the Cloudera system.
Step 2: Each sentences in the positive sentences, do repeat:
Step 3: OneOne-DimensionalVector := transforming one English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera Figure 8
Step 4: Return OneOne-DimensionalVector ;
End;

ALGORITHM 16: implementing the Hadoop Reduce phase of transferring the positive sentences of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) in the distributed system

---

Input: one one-dimensional vector of the positive sentences of the training data set
Output: the positive one-dimensional vectors, called the positive vector group (corresponding to the positive sentences of the training data set)
Begin
Step 1: Receive one one-dimensional vector;
Step 2: Add this one-dimensional vector into PositiveVectorGroup;
Step 3: Return PositiveVectorGroup - the positive one-dimensional vectors, called the positive vector group (corresponding to the positive sentences of the training data set);
End;

ALGORITHM 17: performing the Hadoop Map phase of transferring the negative sentences of the training data set into the negative one-dimensional vectors (called the negative vector group of the training data set) in the distributed system

---

Input: the negative sentences of the training data set
Output: one one-dimensional vector of the negative sentences of the training data set
Begin
Step 1: Input the negative sentences into the Hadoop Map in the Cloudera system.
Step 2: Each sentences in the negative sentences, do repeat:
Step 3: OneOne-DimensionalVector := transforming one English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera Figure 8
Step 4: Return OneOne-DimensionalVector ;
End;

ALGORITHM 18: implementing the Hadoop Reduce phase of transferring the negative sentences of the training data set into the negative multi-dimensional vectors (called the negative vector group of the training data set) in the distributed system

---

Input: one one-dimensional vector of the negative sentences of the training data set

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

ALGORITHM 19: performing the Hadoop Map phase of using the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the parallel environment

---

Input: one one-dimensional vector of a document in the testing data set; the positive vector group and the negative vector group of the training data set.
Output: the result of clustering the vector into either the positive vector group or the negative vector group.
Begin
Step 1: Input this vector, the positive vector group and the negative vector group into the Hadoop Map in the Cloudera system.
Step 2: Select randomly k centres (centroid) of k clusters. Each cluster is represented by the centre of this cluster.
Step 3: Calculate the distance between objects to k centres using Euclidean distance.
Step 4: Group objects into the closest group.
Step 5: Identify the new centre of the clusters.
Step 6: Repeat step 2 until no object groups change.
Step 7: Return the result of clustering the vector into either the positive vector group or the negative vector group;// the output of the Hadoop Map
End;

ALGORITHM 20: implementing the Hadoop Reduce phase of using the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the parallel environment

---

Input: the result of clustering the vector into either the positive vector group or the negative vector group – the output of the Hadoop Map
Output: the result of clustering the vector into either the positive vector group or the negative vector group.
Begin
Step 1: Receive the result of clustering the vector into either the positive vector group or the negative vector group;
Step 2: Return the result of clustering the vector into either the positive vector group or the negative vector group;
End;

ALGORITHM 21: performing the Hadoop Map phase of use the BIRCH and the one-dimensional vectors to cluster one document of the testing data set into either the positive or the negative in the distributed environment

---

Input: one document of the testing data set; the positive vector group and the negative vector group of the training data set.
Output: the result of the sentiment classification of one one-dimensional vector (corresponding to one sentence of this document) into either the positive vector group or the negative vector group
Begin

Step 1: Input this document, the positive vector group and the negative vector group into the Hadoop Map in the Cloudera system.
Step 2: TheOne-dimensionalVectors := transferring all the sentences of one document of the testing data set into the one-dimensional vectors of the document of testing data set based on the sentiment lexicons of the bESD in the parallel network environment Figure 9;
Step 3: Each one-dimensional vector in TheOne-dimensionalVectors, do repeat:
Step 4: OneResult := using the BIRCH to cluster one one-dimensional vector (corresponding one sentence of one document of the testing data set) into either the positive vector group or the negative vector group of the training data set in the parallel environment Figure 12;
Step 5: Return OneResult; // the output of the Hadoop Map
End;

ALGORITHM 22: performing the Hadoop Reduce phase of use the BIRCH and the one-dimensional vectors to cluster one document of the testing data set into either the positive or the negative in the distributed environment

---

Input: OneResult - the result of the sentiment classification of one one-dimensional vector (corresponding to one sentence of this document) into either the positive vector group or the negative vector group
Output: the result of the sentiment classification of this document.
Begin
Step 1: Receive OneResult - the result of the sentiment classification of one one-dimensional vector (corresponding to one sentence of this document) into either the positive vector group or the negative vector group;
Step 2: Add OneResult into the result of the sentiment classification of this document;
Step 3: Return the result of the sentiment classification of this document;
End;

ALGORITHM 23: implementing the Hadoop Map phase of clustering the documents of the testing data set into either the positive or the negative in the sequential environment

---

Input: the documents of the testing data set and the training data set
Output: the result of the sentiment classification of one document of the testing data set;
Begin
Step 1: Transferring the positive sentences of the training data set into the positive one-dimensional vectors (called the positive vector group of the training data set) in the distributed system Figure 10
Step 2: Transferring the negative sentences of the training data set into the negative one-dimensional vectors (called the negative vector group of the training data set) in the distributed system Figure 11
Step 3: Input the documents of the testing data set, the positive vector group and the negative vector group into the Hadoop Map in the Cloudera system
Step 4: Each document in the documents of the testing data set, do repeat:
Step 5: OneResult := Using the BIRCH and the one-dimensional vectors to cluster one document of the testing data set into either the positive or the negative in the distributed environment Figure 13 with the input is this document, the positive vector group and the negative vector group.
Step 6: Return OneResult - the result of the sentiment classification of one document of the testing data set;//the output of the Hadoop Map
End;

ALGORITHM 24: performing the Hadoop Reduce phase of clustering the documents of the testing data set into either the positive or the negative in the parallel environment

---

Input: OneResult - the result of the sentiment classification of one document of the testing data set;//the output of the Hadoop Map
Output: the results of the sentiment classification of the documents of the testing data set;

Begin
Step 1: Receive OneResult ;
Step 2: Add OneResult into the results of the sentiment classification of the documents of the testing data set;
Step 3: Return the results of the sentiment classification of the documents of the testing data set;
End;

AUTHORS

Dr. Vo Ngoc Phu

Current position, grades: researcher; lecturer; Dr of computer science
University studies: Duy Tan University
Scientific interest: artificial intelligence; intelligent systems; expert systems; data mining; machine learning; distributed network environments; natural language processing;
Publications: 19 ISI manuscripts and 2 conferences
Experience: 10 years of researching; teaching

Dr. Vo Thi Ngoc Tran

Current position, grades: researcher; lecturer; Dr of computer science
University studies: Ho Chi Minh City University of Technology
Scientific interest: artificial intelligence; intelligent systems; expert systems; data mining; machine learning; distributed network environments; natural language processing;
Publications: 19 ISI manuscripts and 1 conferences
Experience: 10 years of researching; teaching