ENGLISH SENTIMENT CLASSIFICATION USING A BIRCH ALGORITHM AND THE SENTIMENT LEXICONS-BASED ONE-DIMENSIONAL VECTORS OF A GOWER-2 COEFFICIENT

1DR.VO NGOC PHU, 2VO THI NGOC TRAN
1Nguyen Tat Thanh University, 300A Nguyen Tat Thanh Street, Ward 13, District 4, Ho Chi Minh City, 702000, Vietnam
2School of Industrial Management (SIM), Ho Chi Minh City University of Technology - HCMUT, Vietnam National University, Ho Chi Minh City, Vietnam
E-mail: 1vongocphu03hca@gmail.com, vongocphu@ntt.edu.vn, 2vtntran@HCMUT.edu.vn

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 Iterative 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 English testing data set, which is 8,500,000 documents including the 4,250,000 positive and the 4,250,000 negative based on our English training data set which is 5,000,000 sentences comprising the 2,500,000 positive and the 2,500,000 negative. We calculate the sentiment scores of English terms (verbs, nouns, adjectives, adverbs, etc.) by using a GOWER-2 coefficient (G2C) 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 modeling (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 2,500,000 positive or the 2,500,000 negative of the training data set. We tested the proposed model in both a sequential environment and a distributed network system. We achieved 87.82% 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.

Keywords: English Sentiment Classification; Distributed System; Parallel System; GOWER-2 Similarity Coefficient; Cloudera; Hadoop Map And Hadoop Reduce; Clustering Technology; Balanced Iterative Reducing And Clustering Using Hierarchies Algorithm.

1. INTRODUCTION

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 GOWER-2 similarity measure (G2C) and a Balanced Interactive 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 GOWER-2 similarity measure (G2C) 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 modeling 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 GOWER-2 coefficient (G2C) 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.

The aim of this study is as follows:
1) Many surveys and commercial applications can use the results of this work in a significant way.
2) The algorithms are built in the proposed model.
3) This survey can certainly be applied to other languages easily.
4) The results of this study can significantly be applied to the types of other words in English.
5) Many crucial contributions are listed in the Future Work section.
6) The algorithm of data mining is applicable to semantic analysis of natural language processing.
7) This study also proves that different fields of scientific research can be related in many ways.
8) Millions of English documents are successfully processed for emotional analysis.
The semantic classification is implemented in the parallel network environment.

The principles are proposed in the research.

The Cloudera distributed environment is used in this study.

The proposed work can be applied to other distributed systems.

This survey uses Hadoop Map (M) and Hadoop Reduce (R).

Our proposed model can be applied to many different parallel network environments such as a Cloudera system.

This study can be applied to many different distributed functions such as Hadoop Map (M) and Hadoop Reduce (R).

The BIRCH – related Algorithm are proposed in this survey.

The G2C – 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 modeling (VSM), GOWER-2 similarity measure (G2C), Balanced Iterative 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 (Pointwise Mutual Information) equation and SO (Sentiment Orientation) equation are used for determining polarity of one word (or one phrase), and strength of sentiment orientation of this word (or this phrase). Jaccard measure (JM) is also used for calculating polarity of one word and the equations from this Jaccard measure are also used for calculating strength of sentiment orientation this word in other research. PMI, Jaccard, Cosine, Ochiai, Tanimoto, and Sorensen measure are the similarity measure between two words; from those, we prove that the GOWER-2 coefficient (G2C) 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 generated several Norwegian sentiment lexicons by extracting sentiment information from two different types of Norwegian text corpus, namely, news corpus and discussion forums. The methodology was based on the Pointwise Mutual Information (PMI). The authors introduced a modification of the PMI that considered small "blocks" of the text instead of the text as a whole. The study in [2] introduced a simple algorithm for unsupervised learning of semantic orientation from extremely large corpora, etc.

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 was measurement, which is the fundamental task for word polarity classification. The research in [15] proposed a new method to estimate impression of short sentences considering adjectives. In the proposed system, first, an input sentence was analyzed and preprocessed to obtain keywords. Next, adjectives were taken out from the data which was 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] investigated the problem of sentiment analysis of the online review. In the study [17], the authors were 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 to the similarity coefficients to calculate the valences of words are in [28-32]. The English dictionaries are [33-38] and there are more than 55,000 English words (including English nouns, English adjectives, English verbs, etc.) from them.

The studies related to the Balanced Iterative Reducing and Clustering using Hierarchies algorithm (BIRCH) are in [39-44]. The authors in [39] evaluated 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 was proposed, based on a new in-memory data structure called CF-tree, which served 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 GOWER-2 coefficient (G2C) 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 modeling (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 considered multi-label text classification task and apply various feature sets. The authors considered a subset of multi-labeled files from the Reuters-21578 corpus. The authors used 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] introduced a new weighting method based on statistical estimation of the importance of a word for a specific categorization problem. This method also had the benefit to make feature selection implicit, since useless features for the categorization problem get a very small weight.

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

3. DATA SET

In Fig. 1 below, the testing data set includes 8,500,000 documents in the movie field, which contains 4,250,000 positive documents and 4,250,000 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 labeled positive and negative for them.

In Fig. 2 below, the training data set includes 5,000,000 sentences in the movie field, which contains 2,500,000 positive sentences and 2,500,000 negative sentences in English. All the sentences in our English training data set are automatically extracted from English Facebook, English websites and social networks; then we labeled positive and negative for them.

4. METHODOLOGY

This section comprises two parts: In the first part, we create the sentiment lexicons in English in both a sequential environment and a distributed system in the sub-section (4.1). In the second part, we 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: In the first sub-section of this section, we identify a sentiment value of one word (or one phrase) in English in the sub-section (4.1.1). In the second part of this section, we create a basis English sentiment dictionary (bESD) in a sequential system in the sub-section (4.1.2). In the third sub-section of this section, we 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 comprises two parts: In the first part of this section, we 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). In the second part of this
section, we 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: In the first sub-section of this section, we identify a sentiment value of one word (or one phrase) in English in the sub-section (4.1.1). In the second part of this section, we create a basis English sentiment dictionary (bESD) in a sequential system in the sub-section (4.1.2). In the third sub-section of this section, we 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 G2C through a Google search engine with AND operator and OR operator, as the following diagram in Fig. 3 below shows.

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

\[
PMI(w_i, w_j) = \log_2 \left( \frac{P(w_i, w_j)}{P(w_i) \cdot P(w_j)} \right)
\]  

(1)

and SO (sentiment orientation) of word \( w_i \) has the equation

\[
SO(w_i) = PMI(w_i, positive) - PMI(w_i, negative)
\]  

(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 \( w_i \) and \( w_j \) has the equations

\[
Jaccard(w_i, w_j) = \frac{|w_i \cap w_j|}{|w_i \cup w_j|}
\]  

(3)

and other type of the Jaccard equation between two words \( w_i \) and \( w_j \) has the equation

\[
Jaccard(w_i, w_j) = J(w_i, w_j) = \frac{F(w_i, w_j)}{F(w_i) + F(w_j) - F(w_i, w_j)}
\]  

(4)

and SO (sentiment orientation) of word \( w_i \) has the equation

\[
SO(w_i) = \sum \text{Sim}(w_i, positive) - \sum \text{Sim}(w_i, positive)
\]  

(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 English words. 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 the information as follows: 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 GOWER-2 coefficient (G2C) are the similarity measures between two words, and they can perform the same functions and with the same characteristics; so G2C is used in calculating the valence of the words. In addition, we prove that G2C can be used in identifying the valence of the English word through the Google search engine with the AND operator and OR operator.

With the GOWER-2 coefficient (G2C) in [45-50], we have the equation of the G2C as follows:

\[
GOWER = 2 \text{Coefficient}(a, b) = \frac{A6}{B6} \quad (6)
\]

with \(a\) and \(b\) are the vectors.\n
\[
A6 = (a \cap b) * (\neg(a \cap \neg b))
\]

\[
B6 = ([a \cap b] + (\neg a \cap b)) * ([a \cap b] + (a \cap \neg b)) * ([\neg a \cap b] + (\neg a \cap \neg b)) *
\]

From the eq. (1), (2), (3), (4), (5), (6), we propose many new equations of the G2C 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 \(w2\).

\[
\text{Gower} - 2 \text{Coefficient}(w1, w2) = GOWER - 2 \text{Coefficient}(w1, w2) = \frac{G2C(w1, w2) * P(\neg w1, \neg w2)}{A7} \quad (7)
\]

with

\[
A7 = [P(w1, w2) + P(\neg w1, \neg w2)] \quad (8)
\]

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

\[
G2C(w, \text{positive_query}) = \frac{P(w, \text{positive_query}) * P(\neg w, \neg \text{positive_query})}{A9} \quad (9)
\]

with

\[
A9 = [P(w, \text{positive_query}) \quad + P(\neg w, \neg \text{positive_query}) \quad + P(w, \neg \text{positive_query}) \quad + P(\neg w, \text{positive_query}) ] \quad (10)
\]

In eq. (7), \(w1\) is replaced by \(w\) and \(w2\) is replaced by negative_query. We have eq. (10). Eq. (10) is as follows:
G2C \( (w, \text{negative\_query}) \)
= \( \frac{P(w, \text{negative\_query}) \times P(\neg w, \neg \text{negative\_query})}{A10} \) (10)

with

\[
A10 = [ P(w, \text{negative\_query}) \\
+ P(\neg w, \text{negative\_query}) ] \\
+ [ P(w, \text{negative\_query}) \\
+ P(w, \neg \text{negative\_query}) ] \\
+ [ P(\neg w, \text{negative\_query}) \\
+ P(\neg w, \neg \text{negative\_query}) ] \\
+ [ P(w, \neg \text{negative\_query}) \\
+ P(\neg w, \neg \text{negative\_query}) ]
\]

with:

- \( w, w1, w2 \): are the English words (or the English phrases)
- \( P(w1, w2) \): number of returned results in Google search by keyword \((w1 \text{ and } w2)\). We use the Google Search API to get the number of returned results in search online Google by keyword \((w1 \text{ and } w2)\).
- \( P(w1) \): 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 \(w1\).
- \( P(w2) \): 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 \(w2\).
- \( \text{Valence}(W) = \text{SO\_G2C}(w) \): valence of English word (or English phrase) \(w\); is SO of word (or phrase) by using the GOWER-2 coefficient (G2C)
- \( \text{positive\_query} \): \{ 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.
- \( \text{negative\_query} \): \{ 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.
- \( P(w, \text{positive\_query}) \): 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 \(w\)).
- \( P(w, \text{negative\_query}) \): 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 \(w\)).
- \( P(\neg w, \text{positive\_query}) \): 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)).
- \( P(\neg w, \text{negative\_query}) \): 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)).
- \( P(\neg w, \neg \text{positive\_query}) \): number of returned results in Google search by keyword ((not \(w\) 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)))).
- \( P(\neg w, \neg \text{negative\_query}) \): number of returned results in Google search by keyword ((not \(w\) 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 G2C \((w,\text{positive\_query})\) is as equal as 1.The English word \(w\) is the farthest of positive\_query if G2C \((w,\text{positive\_query})\) is as equal as 0. The English word \(w\) belongs to positive\_query being the positive group of the English words if G2C \((w,\text{positive\_query}) > 0 \text{ and } \text{G2C}(w,\text{positive\_query}) \leq 1\).The English word \(w\) is the nearest of negative\_query if G2C \((w,\text{negative\_query})\) is as equal as 1. The English word \(w\) is the farthest of negative\_query if G2C \((w,\text{negative\_query})\) is as equal as 0. The English word \(w\) belongs to negative\_query being the negative group of the English words if G2C \((w,\text{negative\_query}) > 0 \text{ and } \text{G2C}(w,\text{negative\_query}) \leq 1\).
G2C(w, negative_query) ≤ 1. So, the valence of the English word w is the value of G2C(w, positive_query) subtracting the value of G2C(w, negative_query) and the eq. (8) is the equation of identifying the valence of the English word w.

We have the information about G2C as follows:

- G2C(w, positive_query) ≥ 0 and G2C(w, positive_query) ≤ 1.
- G2C(w, negative_query) ≥ 0 and G2C(w, negative_query) ≤ 1
- If G2C(w, positive_query) = 0 and G2C(w, negative_query) = 0 then SO_G2C(w) = 0.
- If G2C(w, positive_query) = 1 and G2C(w, negative_query) = 0 then SO_G2C(w) = 1.
- If G2C(w, positive_query) = 0 and G2C(w, negative_query) = 1 then SO_G2C(w) = -1.
- If G2C(w, positive_query) = 1 and G2C(w, negative_query) = 1 then SO_G2C(w) = 0.

So, SO_G2C(w) ≥ -1 and SO_G2C(w) ≤ 1.

The polarity of the English word w is positive polarity if SO_G2C(w) > 0. The polarity of the English word w is negative polarity if SO_G2C(w) < 0. The polarity of the English word w is neutral polarity if SO_G2C(w) = 0. In addition, the semantic value of the English word w is SO_G2C(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 GOWER-2 coefficient (G2C) in [39,40].

### 4.1.1 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 G2C in a sequential system, as the following diagram in Fig. 4 below shows.

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

We proposed the algorithm 1 to perform this section.

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

**Output:** a basis English sentiment dictionary (bESD)

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

**Step 2:** By using eq. (8), eq. (9), and eq. (10) of the calculating a valence of one word for 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 G2C through the Google search engine with AND operator and OR operator.

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

**Step 4:** End Repeat – End Step 1;

**Step 5:** Return bESD;

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

### 4.1.3 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 G2C in a parallel network environment, as the following diagram in Fig. 5 below shows.
In Fig. 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 built the algorithm 2 to implement the Hadoop Map phase.

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

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

**Step 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.
**Step 3:** Return this term.

We proposed the algorithm 3 to perform the Hadoop Reduce phase.

**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 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: In the first part of this section, we 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). In the second part of this section, we 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

**Fig. 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**

**Input:** The testing data set

**Output:** The testing data set is classified into either the positive polarity or the negative polarity.

**Step 1:** The documents

**Step 2:** One document

**Step 3:** The sentences

Transfer the sentences into the one-dimensional vectors based on the sentiment lexicons of the bESD in the sequential environment.

**Step 4:** One one-dimensional vector

Using the BIRCH to cluster one one-dimensional vector into either the positive vector group or the negative vector group in the sequential environment.

**Step 5:** The result of the sentiment classification of this document

**Step 6:** The result of the sentiment classification of the documents of the testing data set.
In Fig. 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.

In Fig. 6, we perform the proposed model in the sequential system: 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 proposed 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. 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 2;
Step 6: Return the one-dimensional vectors of this document;

We built 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.

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 OneOne-dimensionalVector into the positive vector group;
Step 5: End Repeat – End Step 2;
Step 6: Return the positive vector group;

We proposed 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.

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 OneOne-dimensionalVector into the positive vector group;
Step 5: End Repeat – End Step 2;
Step 6: Return the positive vector group;
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 Interactive Reducing and Clustering using Hierarchies algorithm (BIRCH) in [39-44], we build 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 proposed the algorithm 9 to cluster 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
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 built the algorithm 10 to cluster 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;
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

In Fig. 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:

In Fig. 7, we perform the proposed model in the parallel system: 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 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 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.

In Fig. 8, we transfer 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 the output of the Hadoop Map. We built the algorithm 11 to perform the Hadoop Map phase

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 proposed the algorithm 12 to perform the Hadoop Reduce phase:
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
Step 1: Receive one term;
Step 2: Add this term into the one-dimensional vector;
Step 3: Return the one-dimensional vector;

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

We proposed the algorithm 14 to perform the Hadoop Reduce phase
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;

In Fig. 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 comprise two phases: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map is one document of the testing data set. 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 phase, 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-dimensional vectors of this document.

We built the algorithm 13 to perform the Hadoop Map phase
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 transferring one English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera in Fig. 8 with the input is this sentence
Step 5: Return one one-dimensional vector; // the output of the Hadoop Map phase.

In Fig. 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. In Fig. 10, the stage includes two phases: the Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase is the positive 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).
Fig. 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.

We built the algorithm 15 to perform the Hadoop Map phase
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-DimentionalVector := The transferring one English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera in Fig. 7
Step 4: Return OneOne-DimentionalVector;

We proposed the algorithm 16 to implement the Hadoop Reduce phase
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);

In Fig. 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.

Fig. 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.

In Fig. 11, the stage includes two phases: the Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase is the negative 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 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 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 built the algorithm 17 to perform the Hadoop Map phase
Input: the negative sentences of the training data set
Output: one one-dimensional vector 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-DimentionalVector := the transferring one English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera in Fig. 8
Step 4: Return OneOne-DimensionalVector;

We proposed the algorithm 18 to implement the Hadoop Reduce phase
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);

In Fig. 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.

![Fig. 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](image)

In Fig. 12, this stage has two phases: 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 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 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 the result of the sentiment classification of this vector.

We built the algorithm 19 to perform the Hadoop Map phase
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 built the algorithm 20 to implement the Hadoop Reduce phase
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;

In Fig. 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.

We proposed the algorithm 21 to perform the Hadoop Map phase
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 in Fig. 9;
Step 3: Each one-dimensional vector in TheOne-dimensionalVectors do repeat:
Step 4: OneResult := the using 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 in the parallel environment in Fig. 12;
Step 5: Return OneResult; // the output of the Hadoop Map

We built the algorithm 22 to perform the Hadoop Reduce phase
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;

In Fig. 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 built the algorithm 23 to implement the Hadoop Map phase
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: The 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 in Fig. 10;
Step 2: The 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 in Fig. 10;
group of the training data set) in the distributed system in Fig. 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 := The 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 in Fig. 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

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 8,500,000 documents of the testing data set. To implement the proposed model, we have already used the 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: Intel® Server Board S1200V3RPS, Intel® Pentium® Processor G3220 (3M Cache, 3.00 GHz), 2GB G2C3-10600 EG2C 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: Intel® Server Board S1200V3RPS, Intel® Pentium® Processor G3220 (3M Cache, 3.00 GHz), 2GB G2C3-10600 EG2C 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
In this paper, our model has been tested on the 8,500,000 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 87.82% 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 37,142,852 seconds / 8,500,000 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 with 3 nodes which is 11,047,614 seconds / 8,500,000 English documents. The average time of the emotion classification of using the BIRCH and the one-dimensional vectors in the Cloudera parallel network environment with 9 nodes, which is 4,139,204 seconds / 8,500,000 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 with 6 nodes is 6,324,807 seconds / 8,500,000 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 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 methods (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.

### Future Work

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

### REFERENCES:


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


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


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


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


APPENDICES

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

GOWER-2 coefficient (G2C)
Semantic classification, sentiment classification: SC

<table>
<thead>
<tr>
<th>Studies</th>
<th>P M I</th>
<th>J M</th>
<th>Language</th>
<th>SD</th>
<th>DT</th>
<th>G 2 C</th>
<th>SC</th>
<th>Other measures</th>
<th>Search engines</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Yes</td>
<td>No</td>
<td>English</td>
<td>Yes</td>
<td>No</td>
<td>N o</td>
<td>Yes</td>
<td>No</td>
<td>No Mention</td>
</tr>
<tr>
<td>[2]</td>
<td>Yes</td>
<td>No</td>
<td>English</td>
<td>Yes</td>
<td>No</td>
<td>N o</td>
<td>Yes</td>
<td>Late</td>
<td>Semantic Analysis (LSA)</td>
</tr>
<tr>
<td>[3]</td>
<td>Yes</td>
<td>No</td>
<td>English</td>
<td>Yes</td>
<td>No</td>
<td>N o</td>
<td>Yes</td>
<td>Base</td>
<td>line; Turney-inspired; NB; Cluster+ NB; Human</td>
</tr>
<tr>
<td>[4]</td>
<td>Yes</td>
<td>No</td>
<td>English</td>
<td>Yes</td>
<td>No</td>
<td>N o</td>
<td>Yes</td>
<td>SimRank</td>
<td>Google Search engine</td>
</tr>
<tr>
<td>[5]</td>
<td>Yes</td>
<td>No</td>
<td>English</td>
<td>Yes</td>
<td>No</td>
<td>N o</td>
<td>Yes</td>
<td>No Mention</td>
<td>Altavista Search engine</td>
</tr>
<tr>
<td>[6]</td>
<td>Yes</td>
<td>No</td>
<td>English</td>
<td>Yes</td>
<td>No</td>
<td>N o</td>
<td>Yes</td>
<td>No Mention</td>
<td>Google Search engine</td>
</tr>
</tbody>
</table>

[7] Yes | No | Eng|ish| Chine|se| Y es | Y es | N o | Y es | SV M(CN); SV M(EN); SV M(ENC N1); SV M(ENC N2); TSV M(CN); TSV M(ENC N1); TSV M(ENC N2); CoTrain | No Mention |

[8] Yes | No | Eng|lish| Spa|nish| Y es | Y es | N o | Y es | SO Calculation SVM | Google |

[9] Yes | No | Chi|nes|e| Ti|betan| Y es | Y es | N o | Y es | - Feature selection - Expectation Cross | No Mention |
<table>
<thead>
<tr>
<th>Reference</th>
<th>Language</th>
<th>Information Bottleneck Method (IB); LE</th>
<th>Information Gain</th>
<th>Entropy - Information Gain</th>
<th>DF, CHI, MI and IG</th>
<th>MI and IGI</th>
<th>N</th>
<th>D</th>
<th>谷歌搜索</th>
<th>a.</th>
<th>Dice; NGD</th>
<th>No Mention</th>
</tr>
</thead>
<tbody>
<tr>
<td>[10]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>DF, CHI, MI and IG</td>
<td>No Mention</td>
<td>N</td>
<td>Y</td>
<td>Yes</td>
<td>Yes</td>
<td>Dice; NGD</td>
<td>No Mention</td>
</tr>
<tr>
<td>[11]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Altavista</td>
<td>Yes</td>
<td>N</td>
<td>Y</td>
<td>Yes</td>
<td>Yes</td>
<td>Dice; Overlap</td>
<td>Google</td>
</tr>
<tr>
<td>[12]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>SVM</td>
<td>Yes</td>
<td>N</td>
<td>Y</td>
<td>Yes</td>
<td>Yes</td>
<td>No Mention</td>
<td>No Mention</td>
</tr>
<tr>
<td>[13]</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Y</td>
<td>Y</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Google</td>
</tr>
<tr>
<td>[14]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N</td>
<td>Y</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Google</td>
</tr>
<tr>
<td>[15]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N</td>
<td>Y</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Google</td>
</tr>
<tr>
<td>[16]</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N</td>
<td>Y</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No Mention</td>
</tr>
<tr>
<td>[17]</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N</td>
<td>Y</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Google</td>
</tr>
</tbody>
</table>
Table 2: Comparisons of our model’s advantages and disadvantages with the works related to [1-32].

<table>
<thead>
<tr>
<th>Survey No.</th>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| [1]        | Construc
ting sentimen
t 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... |
<table>
<thead>
<tr>
<th>No.</th>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[2]</td>
<td>Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus.</td>
<td>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 G2CKeown (1997). SO-PMI-IR requires a large corpus, but it is simple, easy to implement, unsupervised, and it is not restricted to adjectives.</td>
</tr>
<tr>
<td>[3]</td>
<td>Graph-based user classification</td>
<td>The authors describe several experiments in identifying the 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.</td>
</tr>
<tr>
<td>[4]</td>
<td>A novel, graph-based approach using SimRank.</td>
<td>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 sentiments.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>[5]</td>
<td>Analysis in Twitter for Macedonian</td>
<td>The authors’ experimental results show an F1-score of 92.16, which is very strong and is on par with the best results for English, which were achieved in recent SemEval competitions. In future work, the authors are interested in studying the impact of the raw corpus size, e.g., the authors could only collect half a million tweets for creating lexicons and analyzing 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.</td>
</tr>
<tr>
<td>[6]</td>
<td>Using Web Search Engines for English and Arabic Unsupervised Sentiment Intensity Prediction</td>
<td>- 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.</td>
</tr>
<tr>
<td>[7]</td>
<td>Co-Training for Cross-Lingual Sentiment Classification</td>
<td>The authors propose a co-training approach to making use of unlabeled Chinese data. Experimental results show the effectiveness of the proposed approach, which can outperform the standard inductive classifiers and the transductive classifiers. In future work, the authors will improve the sentiment classification accuracy in the following two ways: 1) The smoothed co-training...</td>
</tr>
<tr>
<td>No.</td>
<td>Categorization</td>
<td>Details</td>
</tr>
<tr>
<td>-----</td>
<td>--------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>8</td>
<td>Cross-Linguistic Sentiment Analysis: From English to Spanish</td>
<td>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.</td>
</tr>
<tr>
<td>9</td>
<td>Microblog Emotion</td>
<td>By emotion orientation analyzing and No Mention</td>
</tr>
<tr>
<td>10</td>
<td>An empirical study of sentiment analysis for Chinese documents</td>
<td>Four feature selection methods (MI, IG, CHI and DF) and five learning methods (centroid classifier, K-nearest neighbor, winnow classifier, Naïve Bayes and SVM) are investigated on a Chinese sentiment corpus with a size of 1021 documents. The experimental results indicate that IG performs the best for sentimental terms selection and SVM exhibits the best performance for sentiment classification. Furthermore, the</td>
</tr>
<tr>
<td>[11]</td>
<td><strong>Adapting Information Bottleneck Method for Automatic Construction of Domain-oriented Sentiment Lexicon</strong></td>
<td>The authors found that sentiment classifiers are severely dependent on domains or topics. 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.</td>
</tr>
<tr>
<td>[12]</td>
<td><strong>Sentiment Classification for Consumer Word-of-Mouth in Chinese: Comparison between Supervised and Unsupervised Approaches</strong></td>
<td>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 exceed the semantic orientation approach when the</td>
</tr>
<tr>
<td>[13]</td>
<td><strong>Modifying SO-PMI for Japanese Weblog Opinion Mining by Using a Balancing Factor and Detecting Neutral Expressions</strong></td>
<td>After these modifications, the authors achieved a well-balanced result: both positive and negative accuracy exceeded 70%. This shows that the authors’ proposed approach not only adapted the SO-PMI for Japanese, but also modified it to analyze Japanese opinions more effectively.</td>
</tr>
<tr>
<td>[14]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[15]</td>
<td><strong>Adjective-Based Estimation of</strong></td>
<td>The adjectives are ranked and top N adjectives are considered as an</td>
</tr>
<tr>
<td>Short Sentence’s Impressions</td>
<td>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) they will improve more in the tasks of keyword extraction and semantic similarity methods to make the proposed system working well with complex inputs.</td>
<td>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).</td>
</tr>
<tr>
<td>Jaccard Index based Clustering Algorithm for Mining Online Review</td>
<td>In this work, the problem of predicting sales performance using sentiment information mined from reviews is studied and a novel JIBCA Algorithm is proposed and mathematically modeled. The outcome of this generates knowledge from mined data that can be useful for forecasting sales.</td>
<td>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.</td>
</tr>
<tr>
<td>Twitter sentiment classification for measuring public health concerns</td>
<td>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 No Mention</td>
<td>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 No Mention</td>
</tr>
<tr>
<td><strong>[20]</strong> Sentiment Classification in Under-Resource d Languages Using Graph-based Semi-supervised Learning Methods</td>
<td>automatically ...</td>
<td>As future work, first, the authors will attempt to use a sophisticated approach to induce better sentiment features. The authors consider such elaborate features improve the classification performance, especially in the book domain. The authors also plan to exploit a much larger amount of unlabeled data to fully take advantage of SSL Algorithm. In particular, the authors found that choosing initially labeled vertices in aG2Cordance 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.</td>
</tr>
<tr>
<td><strong>[22]</strong> Sentiment Classification in Resource-Scarce Languages by using Label Propagation</td>
<td>rich, and useful body of consumer data readily available on Web 2.0.</td>
<td>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 labeled seeds. The authors will exploit the idea of restricting label propagation steps when the available labeled data is quite small.</td>
</tr>
<tr>
<td><strong>[28]</strong> A Vietnamese adjective emotion dictionary based on exploitation of Vietnamese language character</td>
<td>A text-mining approach and combine it with semantic network analysis tools</td>
<td>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 not calculating all Vietnamese words complete-ly; not identifying all Vietnamese adjective phrases. 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, No Mention</td>
</tr>
<tr>
<td><strong>[28]</strong> A Vietnamese adjective emotion dictionary based on exploitation of Vietnamese language character</td>
<td>A text-mining approach and combine it with semantic network analysis tools</td>
<td>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 not calculating all Vietnamese words complete-ly; not identifying all Vietnamese adjective phrases. 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, No Mention</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>istics</strong></td>
<td>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.</td>
<td>It has low accuracy; it misses many sentiment-bearing English words; it misses many sentiment-bearing English phrases because sometimes the valence of an 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.</td>
</tr>
<tr>
<td><strong>[29]</strong> A Valences-Totaling Model for English Sentiment Classification</td>
<td>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.</td>
<td>The results of this work are widely used in applications and researches of the English semantic classification.</td>
</tr>
<tr>
<td><strong>[30]</strong> Shifting Semantic Values of English Phrases for Classification</td>
<td>The authors have used the VTMV to classify 30,000 Vietnamese documents which include the 15,000 positive Vietnamese semantic values (or their sentiment scores) are not changed in any context, then the results of the emotion classification are not high accuracy.</td>
<td>It has a low accuracy</td>
</tr>
</tbody>
</table>
Sentiment Classification

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.

Table 3: Comparisons of our model’s results with the works related to the GOWER-2 coefficient (G2C) in [45-50]

<table>
<thead>
<tr>
<th>Studies</th>
<th>P</th>
<th>M</th>
<th>J</th>
<th>GO WER-2 coefficient (G2C)</th>
<th>Language</th>
<th>SD</th>
<th>DT</th>
<th>Sentiment Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>[45]</td>
<td>Y</td>
<td>e</td>
<td>s</td>
<td>Yes</td>
<td>English</td>
<td>N</td>
<td>M</td>
<td>No mention</td>
</tr>
<tr>
<td>[46]</td>
<td>N</td>
<td>o</td>
<td></td>
<td>Yes</td>
<td>N M</td>
<td>N</td>
<td>M</td>
<td>No mention</td>
</tr>
<tr>
<td>[47]</td>
<td>N</td>
<td>o</td>
<td></td>
<td>Yes</td>
<td>N M</td>
<td>N</td>
<td>M</td>
<td>No mention</td>
</tr>
<tr>
<td>[48]</td>
<td>N</td>
<td>o</td>
<td></td>
<td>Yes</td>
<td>N M</td>
<td>N</td>
<td>M</td>
<td>No mention</td>
</tr>
<tr>
<td>[49]</td>
<td>N</td>
<td>o</td>
<td></td>
<td>Yes</td>
<td>N M</td>
<td>N</td>
<td>M</td>
<td>No mention</td>
</tr>
<tr>
<td>[50]</td>
<td>N</td>
<td>N</td>
<td>Yes</td>
<td></td>
<td>N M</td>
<td>N</td>
<td>N</td>
<td>No</td>
</tr>
</tbody>
</table>

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.

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).
<table>
<thead>
<tr>
<th>Survey</th>
<th>Approach</th>
<th>Benefits</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>[45]</td>
<td>A Survey of Binary Similarity and Distance Measures</td>
<td>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.</td>
<td>No mention</td>
</tr>
<tr>
<td>[46]</td>
<td>Similarity coefficients for classifying relevés</td>
<td>In this study, the clustering procedure of group average sorting was used to construct the dendrogram. It gives an average similarity value within the dendrogram groups. These values can be used to give quantitative definitions to syntaxonomic rank.</td>
<td>No mention</td>
</tr>
<tr>
<td>[47]</td>
<td>Assessment of Similarity Indices for</td>
<td>The purpose of this study is to motivate, describe, and offer</td>
<td>No mention</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Our work</td>
<td>N o</td>
<td>N o</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 4: Comparisons of our model’s benefits and drawbacks with the studies related to the GOWER-2 coefficient (G2C) in [45-50]

<table>
<thead>
<tr>
<th>Survey</th>
<th>Approach</th>
<th>Benefits</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>[48]</td>
<td>Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions</td>
<td>Various distance/similarity measures that are applicable to compare two probability density functions, pdf in short, are reviewed and categorized in both syntactic and semantic relationships. A correlation coefficient and a hierarchical clustering technique are adopted to reveal similarities among numerous distance/similarity measures.</td>
<td>No mention</td>
</tr>
<tr>
<td>[49]</td>
<td>Coefficients Of Association And Similarity, Based On Binary (Presence-Absence) Data: An Evaluation</td>
<td>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.</td>
<td>No mention</td>
</tr>
<tr>
<td>[50]</td>
<td>Finding an appropriate equation to measure the selection of binary similarity and dissimilarity measures</td>
<td></td>
<td>No mention</td>
</tr>
</tbody>
</table>
similarity between binary vectors: case studies on Indonesian and Japanese herbal medicines

multivariate analysis is data dependent. The proposed method can be used to find the most suitable binary similarity and dissimilarity equation wisely for a particular data. Our finding suggests that all four types of matching quantities in the Operational Taxonomic Unit (OTU) table are important to calculate the similarity and dissimilarity coefficients between herbal medicine formulas. Also, the binary similarity and dissimilarity measures that include the negative match quantity $d$ achieve better capability to separate herbal medicine pairs compared to equations that exclude $d$.

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 documents in the testing data set.

<table>
<thead>
<tr>
<th></th>
<th>Testing Dataset</th>
<th>Correct Classification</th>
<th>Incorrect Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative</td>
<td>4,250,000</td>
<td>3,732,176</td>
<td>517,824</td>
</tr>
<tr>
<td>Positive</td>
<td>4,250,000</td>
<td>3,732,524</td>
<td>517,476</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Proposed Model</th>
<th>Class</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our novel model</td>
<td>Negative</td>
<td>87.82%</td>
</tr>
<tr>
<td></td>
<td>Positive</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Average time of the classification /8,500,000 documents.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The GOWER-2 coefficient (G2C) in the sequential environment</td>
</tr>
<tr>
<td>The GOWER-2 coefficient (G2C) in the Cloudera distributed system with 3 nodes</td>
</tr>
<tr>
<td>The GOWER-2 coefficient (G2C) in the Cloudera distributed system with 6 nodes</td>
</tr>
<tr>
<td>The GOWER-2 coefficient (G2C) in the Cloudera distributed system with 9 nodes</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Studies</th>
<th>G2</th>
<th>C</th>
<th>T</th>
<th>Sentiment Classification</th>
<th>PNS</th>
<th>S</th>
<th>D</th>
<th>Language</th>
<th>VSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>[39]</td>
<td>No</td>
<td>N</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>N</td>
<td>No</td>
<td>EL</td>
<td>Yes</td>
</tr>
<tr>
<td>[40]</td>
<td>No</td>
<td>N</td>
<td>Yes</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>EL</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Our work

<table>
<thead>
<tr>
<th>Researches</th>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>[39]</td>
<td>BIRCH: A New Data Clustering Algorithm and Its Applications</td>
<td>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 constraints). 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 &quot;noise&quot; (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.</td>
<td>No mention</td>
</tr>
<tr>
<td>[40]</td>
<td>BIRCH: A New Data Clustering Algorithm and Its Applications</td>
<td>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</td>
<td>No mention</td>
</tr>
</tbody>
</table>

Table 9: Comparisons of our model's advantages and disadvantages with the works in [39-44]
<table>
<thead>
<tr>
<th>No.</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN N and Its Applications</td>
<td>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 points (astronomy), 3D points (biology), 5D points (earth science) and 2D polygons (geography) are presented, demonstrating the applicability of GDBSCAN to real-world problems.</td>
</tr>
<tr>
<td>42</td>
<td>Leaders–Subleaders: As an example, a two level</td>
<td>No mention</td>
</tr>
<tr>
<td>43</td>
<td>Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN N and Its Applications</td>
<td>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 points (astronomy), 3D points (biology), 5D points (earth science) and 2D polygons (geography) are presented.</td>
</tr>
</tbody>
</table>

An efficient hierarchical clustering algorithm for large data sets. 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.
demonstrating the applicability of GDBSCAN to real-world problems.

[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 scales much 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.

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]

<table>
<thead>
<tr>
<th>Studies</th>
<th>G2C</th>
<th>CT</th>
<th>Sentiment Classification</th>
<th>PN</th>
<th>SD</th>
<th>DT</th>
<th>Language</th>
<th>VSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>[51]</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>EL</td>
<td>Yes</td>
</tr>
<tr>
<td>[52]</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>EL</td>
<td>Yes</td>
</tr>
<tr>
<td>[53]</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>EL</td>
<td>Yes</td>
</tr>
<tr>
<td>Our work</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>EL</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Researches</th>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| [51]       | Examining the vector space model, an information retrieval technique and its counterparts | 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 drawba | The drawbacks are that the system yields no theoreti
variation bare fact that each system has its own strengths and weaknesses. What we have sorted out in the authors’ work for vector space modeling is that the model is easy to understand and cheaper to implement, considering the fact that the system should be cost effective (i.e., should follow the space/time constraint. It is also very popular. Although the system has all these properties, it is facing some major drawbacks.

<table>
<thead>
<tr>
<th>Variation</th>
<th>bare fact that each system has its own strengths and weaknesses. What we have sorted out in the authors’ work for vector space modeling is that the model is easy to understand and cheaper to implement, considering the fact that the system should be cost effective (i.e., should follow the space/time constraint. It is also very popular. Although the system has all these properties, it is facing some major drawbacks.</th>
</tr>
</thead>
</table>

cal findings. Weight is associated with the vectors are very arbitrary, and this system is an independent system, thus requiring separate attention. Though it is a promising technique, the current level of success of the vector space model techniques used for information retrieval are not able to satisfy user needs and need extensive attention.

subset of multi-labeled files of the Reuters-21578 corpus. The authors use traditional TF-IDF values of the features and tried both considering and ignoring stop words. The authors also tried several combinations of features, like bi-grams and uni-grams. The authors also experimented with adding LDA results into vector space models as new features. These last experiments obtained the best results.

| 5265 | In this work, the authors consider multi-label text classification tasks and apply various feature sets. The authors consider a subset of multi-labeled files of the Reuters-21578 corpus. The authors use traditional TF-IDF values of the features and tried both considering and ignoring stop words. The authors also tried several combinations of features, like bi-grams and uni-grams. The authors also experimented with adding LDA results into vector space models as new features. These last experiments obtained the best results. |
| 5265 | [52] Latent Dirichlet allocation (LDA). Multi-label text classification. In this work, the authors consider multi-label text classification tasks and apply various feature sets. The authors consider a subset of multi-labeled files of the Reuters-21578 corpus. The authors use traditional TF-IDF values of the features and tried both considering and ignoring stop words. The authors also tried several combinations of features, like bi-grams and uni-grams. The authors also experimented with adding LDA results into vector space models as new features. These last experiments obtained the best results. |

5265 | In this work, the authors consider multi-label text classification tasks and apply various feature sets. The authors consider a subset of multi-labeled files of the Reuters-21578 corpus. The authors use traditional TF-IDF values of the features and tried both considering and ignoring stop words. The authors also tried several combinations of features, like bi-grams and uni-grams. The authors also experimented with adding LDA results into vector space models as new features. These last experiments obtained the best results. |
| 5265 | [53] The K-Nearest Neighbors algorithm for English sentiment classification in the Cloudera distributed system. Despite positive results in some settings, GainRatio failed to show that supervised weighting methods are generally higher than unsupervised ones. The authors believe that Conf-conf is a promising supervised weighting technique. |
ue 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]

<table>
<thead>
<tr>
<th>Studies</th>
<th>Approach</th>
<th>Positives</th>
<th>Negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>[54]</td>
<td>The Machine Learning Approaches Applied to Sentiment Analysis-Based Applications</td>
<td>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.</td>
<td>No mention</td>
</tr>
</tbody>
</table>

| Our work    | Y es                        | Y es                           | N o                        | N o                        | Y es                        | Y es                        | Y es                        | N o                        | Y es                        | Y es                        |

Table 13: Comparisons of our model’s positives and negatives the latest sentiment classification models

No mention

No mention
<p>| [5][6] | Exploiting New Sentiment-Based Meta-Level Features for Effective Sentiment Analysis | Experiments performed with a substantial number of datasets (nineteen) demonstrate that the effectiveness of the proposed sentiment-based meta-level features is not only superior to the traditional bag-of-words representation (by up to 16%) but also is also superior in most cases to state-of-the-art meta-level features previously proposed in the literature for text classification tasks that do not take into account any idiosyncrasies of sentiment analysis. The authors’ proposal is also largely superior to the best lexicon-based methods as well as to supervised combinations of them. In fact, the proposed approach is the only one to produce the best results in all tested datasets in all scenarios. | A line of future research would be to explore the authors’ meta features with other classification algorithms and feature selection techniques in different sentiment analysis tasks such as scoring movies or products. According to their related reviews, 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. |
| [5][7] | 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. | No mention |
| [5][8] | The Combination of Term-Counting Method and Enhanced Contextual Valence Shifters Method | The authors have explored different methods of improving the accuracy of sentiment classification. The sentiment orientation of a document can be positive (+), negative (-), or neutral (0). The authors combine five dictionaries into a new one with 21,137 entries. The new dictionary has many verbs, adverbs, phrases and idioms that were not in five dictionaries before. The study shows that the authors’ proposed method based on the combination of Term-Counting method and Enhanced Contextual Valence Shifters method has improved the accuracy of sentiment classification. The combined method has accuracy 68.984% on the testing dataset and 69.224% on the training dataset. All of these methods are implemented to classify the reviews based on our new dictionary and the Internet Movie Database data set. | No mention |
| [5][9] | Naive Bayes Model with N-GRAM Method, Negation Handling Method, | 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 features. | No mention |</p>
<table>
<thead>
<tr>
<th>Chi-Square Method and Good-Turing Discounting, etc.</th>
<th>thresholds of Good-Turing Discounting method and different minimum frequencies of Chi-Square method to improve the accuracy of sentiment classification.</th>
</tr>
</thead>
</table>

**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.