**English Sentiment Classification using A Hamann Coefficient and a Genetic Algorithm with a Roulette-Wheel Selection in a Parallel Network Environment**

1Vo Ngoc Phu and 2Vo Thi Ngoc Tran  
1Nguyen Tat Thanh University, 300A Nguyen Tat Thanh Street, Ward 13, District 4, 702000 Ho Chi Minh City, Vietnam  
2School of Industrial Management (SIM), Ho Chi Minh City University of Technology-HCMUT, Vietnam National University, Ho Chi Minh City, Vietnam

**Abstract:** We have already studied a data mining field and a natural language processing field for many years. There are many significant relationships between the data mining and the natural language processing. Sentiment classification has had many crucial contributions to many different fields in everyday life such as in political activities, commodity production and commercial activities. A new model using a Hamann Coefficient (HC) and a Genetic Algorithm (GA) with a Fitness Fuction (FF) which is a Roulette-Wheel Selection (RWS) has been proposed for the sentiment classification. This can be applied to a big data. The GA can process many bit arrays. Thus, it saves a lot of storage spaces. We do not need lots of storage spaces to store a big data. Firstly, we create many sentiment lexicons of our basis English Sentiment Dictionary (bESD) by using the HC through a Google search engine with AND operator and OR operator. Next, According to the sentiment lexicons of the bESD, we encode 7,000,000 sentences of our training data set including the 3,500,000 negative and the 3,500,000 positive in English successfully into the bit arrays in a small storage space. We also encrypt all sentences of 9,000,000 documents of our testing data set comprising the 4,500,000 positive and the 4,500,000 negative in English successfully into the bit arrays in the small storage space. We use the GA with the RWS to cluster one bit array (corresponding to one sentence) of one document of the testing data set into either the bit arrays of the negative sentences or the bit arrays of the positive sentences of the training data set. The sentiment classification of one document is based on the results of the sentiment classification of the sentences of this document of the testing data set. We tested the proposed model in both a sequential environment and a distributed network system. We achieved 88.02% 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 study can be widely used in applications and research of the English sentiment classification.

**Key words:** English sentiment classification, distributed system, Hamann similarity coefficient, Cloudera, Hadoop Map and Hadoop Reduce, Genetic algorithm, Roulette-Wheel selection

**INTRODUCTION**

Many machine-learning methods or methods based on lexicons or a combination of both have been studied for sentiment classification for many years. Sentiment analysis has a wide range of applications in the fields of business, organizations, governments and individuals. About many clustering technologies of a data mining field, a set of objects is processed into classes of similar objects, call clustering data. A set of data objects are similar to each other, called one cluster and the data objects are not similar to objects in other clusters. A number of data clusters can be clustered which can be identified following experience or can be automatically identified as part of clustering method.

A Genetic Algorithm (GA) is a technology of the data mining which is a metaheuristic inspired by the process of natural selection that belongs to the larger class of Evolutionary Algorithms (EA). The GA is commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection.

The genetic algorithm differs from a classical, derivative-based, optimization algorithm in two main ways as summarized as follows: Genetic algorithm: generates a

**Corresponding Author:** Vo Ngoc Phu, Nguyen Tat Thanh University, 300A Nguyen Tat Thanh Street, Ward 13, District 4, 702000 Ho Chi Minh City, Vietnam

2055
population of points at each iteration. The best point in the population approaches an optimal solution. Selects the next population by computation which uses random number generators. Classical algorithm: generates a single point at each iteration. The sequence of points approaches an optimal solution. Selects the next point in the sequence by a deterministic computation.

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

The motivation of this new model is as follows: many algorithms in the data mining field can be applied to natural language processing, specifically semantic classification for processing millions of English documents. A Hamann similarity measure (HC) and a Genetic Algorithm (GA) of the clustering technologies of the data mining filed can be applied to the sentiment classification in both a sequential environment and a parallel network system. This will result in many discoveries in scientific research, hence, the motivation for this study.

The novelty of the proposed approach is as follows: the Hamann similarity measure (HC) and the GA are applied to sentiment analysis. This 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.

To get higher accuracy of the results of the sentiment classification to shorten execution times of the sentiment classification and to save lots of storage spaces, we use the GA with a Fitness Function (FF) which is a Roulette-Wheel Selection (RWS) because as known, the GA processes many bit arrays and the bit arrays always take many small spaces to be run and saved. Unsurprisingly, a storage space of the bit arrays of the training data set is much less than a storage space of all the sentences of the training data set. The HA is used to identify many sentiment values and polarities of many sentiment lexicons of our basis English Sentiment Dictionary (bSSD) through a Google search engine with AND operator and OR operator.

We perform the proposed model as follows: firstly, the valences and the polarities of the sentiment lexicons of the bSSD are identified by using the HA through the Google search engine with AND operator and OR operator. We label all the sentiment lexicons of the bSSD by using many binary bits. Therefore, each term (meaningful word or meaningful phrase) in the sentiment lexicons are shown by one bit array. This bit array provides the information of this term about a content of this term (example as "good", "bad", "very", etc.), a valence of the term. Next, we encrypt all the sentences of the training data set to the bit arrays which are stored in a small storage space. All the positive sentences of the training data set are encoded to the positive bit arrays, called the positive bit array group. All the negative sentences of the training data set are encrypted to the negative bit arrays, called the negative bit array group. All the sentences of one document of the testing data set are encoded to the bit arrays of this document. We use the GA with RWS to cluster one bit array (corresponding to one sentence) of one document of the testing data set into either the positive bit array or the negative bit array of the training data set. This document is clustered into the positive polarity if the number of the bit arrays (corresponding to the sentences) clustered into the positive is greater than the number of the bit arrays (corresponding to the sentences) clustered into the negative in the document. This document is clustered into the negative polarity if the number of the bit arrays (corresponding to the sentences) clustered into the positive is less than the number of the bit arrays (corresponding to the sentences) clustered into the negative in the document. This document is clustered into the neutral polarity if the number of the bit arrays (corresponding to the sentences) clustered into the positive is as equal as the number of the bit arrays (corresponding to the sentences) clustered into the negative in the document. Finally, the sentiment classification of all the documents of the testing data set is implemented completely.

All the above things are firstly implemented in a sequential environment to get an accuracy and an execution time of the proposed model. Then, all the above things are performed in a parallel network system to get the accuracy and the execution times of our proposed model with a purpose which is to shorten the execution times of the model. Many significant contributions of our new model can be applied to many areas of research as well as commercial applications as follows:

- Many surveys and commercial applications can use the results of this study in a significant way
- The algorithms are built in the proposed model
- This survey can certainly be applied to other languages easily
- The results of this study can significantly be applied to the types of other words in English
- Many crucial contributions are listed in the future study
- The algorithm of data mining is applicable to semantic analysis of natural language processing
- This study also proves that different fields of scientific research can be related in many ways.
Two studies related to the PMI measure and Jaccard measure are by Feng et al. (2013). In the survey of Feng et al. (2013), the researchers empirically evaluate the performance of different corpora in sentiment similarity measurement which is the fundamental task for word polarity classification. The research by Nguyen and Hagiwara proposes a new method to estimate impression of short sentences considering adjectives. In the proposed system, first, an input sentence is analyzed and preprocessed to obtain keywords. Next, adjectives are taken out from the data which is queried from Google N-gram corpus using keywords-based templates.

The studies related to the Jaccard measure are by Shikalgar and Dixit (2014), Ji et al. (2015), Omam et al. (2013), Mao et al. (2014), Ren et al. (2014), Netzer et al. (2012) and Ren et al. (2011). The survey by Shikalgar and Dixit (2014) investigates the problem of sentiment analysis of the online review. In the study of Ji et al. (2015), the researchers are addressing the issue of spreading public concern about epidemics. Public concern about a communicable disease can be seen as a problem of its own, etc.

The surveys related the similarity coefficients to calculate the valences of words are by Phu et al. (2017 a-h). The English dictionaries are Anonymous (2017 a-k) and there are more than 55,000 English words (including English nouns, English adjectives, English verbs, etc.) from them.

There are the studies related to the Hamann Coefficient (HC) by Choi et al. (2010), Schraw (1995), David et al. (1959), Sehrrer et al. (2004), Grigsby et al., 1998 and Urbina et al. (2010). The researchers by Choi et al. (2010) collected 76 binary similarity and distance measures used over the last century and reveal their correlations through the hierarchical clustering technique. By Schraw (1995), two ordinal measures of feeling-of-knowing performance appropriate for n × n data arrays are reviewed. Goodman and Kruskal’s gamma provides a measure of association between recognition performance and feeling-of-knowing judgements. The hamann coefficient provides a measure of agreement accuracy. The relative strengths and weaknesses of each measure are compared at length, etc.

The surveys related to the Genetic Algorithm (GA) by Davis (1991), Kora and Krishna (2016), Yang et al. (2016), Esküya and Üzüm, 2016 and Wu et al. (2016). The survey by Davis (1991) sets out to explain what genetic algorithms are and how they can be used to solve real-world problems. In the study Kora and Krishna (2016), Differential Evolution (DE) can be efficiently used to detect the changes in the EEG using optimized features from the EEG beats. For the detection of normal and BBB beats, these DE feature values are given as the input for the LMNN classifier, etc.
There are the researches related to the Roulette-Wheel Selection (RWS) by Panda et al. (2009), Lee et al. (1998), Tat and Tao (2003), Lipowski and Lipowska (2012) and Zou et al. (2006). In the study of Panda et al. (2009), the researchers describe PLANET: a scalable distributed framework for learning tree models over large datasets. The researchers by Lee et al. (1998) present an algorithm for designing 1-D FIR filters using genetic algorithms, etc.

The latest researches of the sentiment classification are Agarwal and Mittal (2016 a-b), Canuto et al. (2016), Ahmed and Dai (2016), Phu and Tuoi (2014), Ngoc et al. (2017), Dat et al. (2017), Phu et al. (2017 a-h). In the research of Agarwal and Mittal (2016 a, b), the researchers present their machine learning experiments with regard to sentiment analysis in blog, review and forum texts found on the World Wide Web and written in English, Dutch and French. The survey by Agarwal and Mittal (2016 a, b) discusses an approach where an exposed stream of tweets from the Twitter microblogging site are pre-processed and classified based on their sentiments. In sentiment classification system the concept of opinion subjectivity has been accounted. In the study, the researchers present opinion detection and organization subsystem which have already been integrated into our larger question-answering system, etc.

The surveys related to the binary code of letters in English are shown by Anonymous (2017a-k). The researches by Anonymous (2017a-k) show all the binary codes of all the letters in English completely.

There are the researches related to transferring a decimal to a binary code by Anonymous (2017 a-k). The surveys by Anonymous (2017a-k) show how to transfer one decimal to one binary code.

**Data set:** All documents and all sentences of our testing data set and training data set were automatically extracted from English Facebook, English websites and social networks. Then, we labeled positive and negative for all the documents of the testing data set and we also labeled positive and negative for all the sentences of the training data set.

The testing data set was built manually and it includes the 9,000,000 documents in the movie field which has the 4,500,000 positive and 4,500,000 negative in English in Fig. 1.

Figure 2 shows, we built the training data set comprising 7,000,000 sentences in the movie field which contains the 3,500,000 positive and the 3,500,000 negative in English.
MATERIALS AND METHODS

There are two parts in this study as follows: the first part is the sub-section which we create the sentiment lexicons in English in both a sequential environment and a distributed system.

The second part is the sub-section which we use the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS). The Fitness Fuction (FF) to cluster the documents of the testing data set into either the positive vector group or the negative vector group in both a sequential environment and a distributed system.

The sentiment lexicons in English: There are three parts in this study: in the first part, we identify a sentiment value of one word (or one phrase) in English. In the 2nd part, we create a basis English Sentiment Dictionary (bESD) in a sequential system. In the 3rd part, we create a basis English Sentiment Dictionary (bESD) in a parallel environment.

A valence of one word (or one phrase) in English: In this part, the valence and the polarity of one English word (or phrase) are calculated by using the HC through a Google search engine with AND operator and OR operator as shown in Fig. 3.

The researchers of the surveys by Bai et al. (2014), Turney and Littman (2002), Malouf and Mullen (2017), Scheible (2010), Jovanoski et al. (2015), Huit et al. (2016), Wan (2009), Brooke et al. (2009), Jiang et al. (2015), Tan and Zhang (2008), Du et al. (2010), Zhang et al. (2010), Wang and Araki (2007) and Feng et al. (2013) use an equation about Pointwise Mutual Information (PMI) between two words w_i and w_j as follows Eq. 1:

\[
\text{PMI}(w_i, w_j) = \log_{2} \left( \frac{p(w_i, w_j)}{p(w_i) \times p(w_j)} \right)
\]  

(1)

and an equation about SO (Sentiment Orientation) of word w_i as follow Eq. 2:

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

(2)

The researchers of the studies by Bai et al. (2014), Turney and Littman (2002), Malouf and Mullen (2017), Scheible (2010), Jovanoski et al. (2015), Huit et al. (2016), Wan (2009) and Brooke et al. (2009) use the positive and the negative of Eq. 2 in English as follows:

positive = {good, nice, excellent, positive, fortunate, correct, superior} and negative = {bad, nasty, poor, negative, unfortunate, wrong, inferior}.

The researchers of the studies by Turney and Littman (2002), Malouf and Mullen (2017) and Jovanoski et al. (2015) use the PMI equations with the AltaVista search engine and the researchers of the researches by Scheible (2010), Huit et al. (2016) and Brooke et al. (2009) use the PMI equations with the Google search engine.

Besides, the researchers of the study by Scheible (2010) also use German, the researchers of the study by Jovanoski et al. (2015) also use Macedonian, the researchers of the survey by Huit et al. (2016) also use Arabic, the researchers of the study by Wan (2009) also use Chinese and the researchers of the research by Brooke et al. (2009) also use Spanish. In addition, the Bing search engine is also used by Huit et al. (2016).

The researchers of the surveys by Jiang et al. (2015), Tan and Zhang (2008), Du et al. (2010) and Zhang et al. (2010) use the PMI equations in Chinese not English and Tibetan is also added by Jiang et al. (2015).

About the search engine, the researchers of the researches by Du et al. (2010) and Zhang et al. (2010) use the AltaVista search engine and the researchers by Zhang et al. (2010) use three search engines, such as the Google search engine, the Yahoo search engine and the Baidu search engine. The survey of Wang and Araki (2007) use the PMI equations in Japanese with the Google search engine. The researchers by Feng et al. (2013) and also use the PMI equations and Jaccard equations with Google search engine in English.
The researchers of the studies by Feng et al. (2013), Shikalgar and Dixit (2014), Ji et al. (2015), Omar et al. (2013), Mao et al. (2014), Ren et al. (2014), Netzer et al. (2012), Ren et al. (2011) use the equations about Jaccard between two words \( w_i \) and \( w_j \) as follows Eq. 3:

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

(3)

and other type of the Jaccard equation between two words \( w_i \) and \( w_j \) has in Eq. 4:

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

(4)

and an equation about SO (Sentiment Orientation) of word \( w_i \) as follows Eq. 5:

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

(5)

The researchers of the surveys by Feng et al. (2013), Shikalgar and Dixit (2014), Ji et al. (2015), Omar et al. (2013), Mao et al. (2014), Ren et al. (2014) and Netzer et al. (2012) use the positive and the negative of Eq. 5 in English as follows: positive = {good, nice, excellent, positive, fortunate, correct, superior} and negative = {bad, nasty, poor, negative, unfortunate, wrong, inferior}.

The studies by Feng et al. (2013) and Ji et al. (2015) use the Jaccard equations with the Google Search engine in English. The surveys by Shikalgar and Dixit (2014) and Netzer et al. (2012) use the Jaccard equations in English. The researchers by Ren et al. (2014) and Ren et al. (2011) use the Jaccard equations in Chinese. The researchers by Omar et al. (2013) use the Jaccard equations in Arabic. The Jaccard equations with the Chinese search engine in Chinese are used by Mao et al. (2014).

The researchers by Phu et al. (2017a-h) 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 researchers by Phu et al. (2017a-h) 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 researchers by Phu et al. (2017a-h) used the Sorensen Coefficient through the Google Search engine with AND operator and OR operator to calculate the sentiment values of the words in English. The researchers by Phu et al. (2017a-h) 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 researchers by Phu et al. (2017a-h) 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.

According to 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.

PMI, Jaccard, Cosine, Ochiai, Sorensen, Tanimoto and Hamann Coefficient (HC) are the similarity measures between two words by Bai et al. (2014), Turney and Littman (2002), Malouf and Mullen (2017), Scheible (2010), Jovanovski et al. (2015), Huait et al. (2016), Wan (2009), Brooke et al. (2009), Jiang et al. (2015), Tan and Zhang (2008), Do et al. (2010), Zhang et al. (2010), Wang and Araki (2007), Feng et al. (2013), Shikalgar and Dixit (2014), Ji et al. (2015), Omar et al. (2013), Mao et al. (2014), Ren et al. (2014), Netzer et al. (2012), Ren et al. (2011), Hernandez-Ugalde et al. (2011), Poromarenko et al. (2002), Meyer et al. (2004), Madenovic et al. (2008), Tamas et al. (2001) and Phu et al. (2017a-h) and they can perform the same functions and with the same characteristics, so, the HC is used in calculating the valence of the words. In addition, we prove that the HC can be used in identifying the valence of the English word through the Google Search with the AND operator and OR operator.

We have an equation about the Hamann Coefficient (HC) by Choi et al. (2010), Schraw (1995), David et al. (1959), Schnyer et al. (2004), Grigsby et al. (1998), Urbina et al. (2010) as follows Eq. 6:

\[
\text{Hamann coefficient}(a, b) = \text{Hamann measure}(a, b) = \frac{[(a \cap b) + (-a \cap -b) - (-a \cap -b) + (a \cap -b)]}{(a \cap b) + (-a \cap b) + (a \cap -b) + (-a \cap -b)}
\]  

(6)

with a and b are the vectors. Based on Eq. 1-6, we propose many new equations of the HC to calculate the valence and the polarity of the English words (or the English phrases) through the Google search engine as the following equations.

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 \( w_i \), and b is replaced by \( w_j \), Eq. 7.
Hamann measure \( w_1, w_2 \) =
\[
\text{Hamann coefficient (} w_1, w_2 \text{)} = \\
\frac{[P(w_1, w_2) + P(-w_1, -w_2)] - [P(-w_1, w_2) + P(w_1, -w_2)]}{[P(w_1, w_2) + P(-w_1, -w_2)] + [P(-w_1, w_2) + P(w_1, -w_2)]}
\]

Equation 7 is similar to 1. Equation 2, 1 is replaced by Eq. 7. We have Eq. 8 as follows:
\[
\text{Valence (} w \text{)} = \frac{SO_{SC(}w)}{\text{HC (} w, \text{positive}_\text{query} \text{)} - \text{HC (} w, \text{negative}_\text{query} \text{)}}
\]

Equation 7 shows \( w_1 \) is replaced by \( w \) and \( w_2 \) is replaced by \( \text{position}_\text{query} \). We have Eq. 9 as follows:
\[
\begin{align*}
A_9 &= [P(w, \text{positive}_\text{query}) + P(-w, \text{negative}_\text{query})] \\
B_9 &= [P(-w, \text{positive}_\text{query}) + P(w, \text{negative}_\text{query})] \\
C_9 &= P(w, \text{positive}_\text{query}) + P(-w, \text{positive}_\text{query}) \\
P(w, \text{negative}_\text{query}) + P(-w, \text{negative}_\text{query})
\end{align*}
\]

Equation 7 shows \( w_1 \) is replaced by \( w \) and \( w_2 \) is replaced by \( \text{negative}_\text{query} \). We have Eq. 10. Equation 10 is as follows:
\[
\text{HC (} w, \text{negative}_\text{query} \text{)} = \frac{A_10 - B_10}{C_10}
\]

Algorithm 1: \( w, w_1, w_2 \) algorithm:
We have the information about \( w, w_1, w_2 \) and etc. as follows:
1) \( w, w_1, w_2 \) are the English words (or the English phrases)
2) \( P(w, w_1) \): number of returned results in Google Search by keyword \( w_1 \) and \( w \). We use the Google Search API to get the number of returned results in search online Google by keyword \( w_1 \) and \( w \).
3) \( P(w) \): number of returned results in Google Search by keyword \( w \). We use the Google Search API to get the number of returned results in search online Google by keyword \( w \).
4) \( P(w) \): number of returned results in Google Search by keyword \( w \). We use the Google Search API to get the number of returned results in search online Google by keyword \( w \).
5) \( \text{Valence}(w) = SO_{SC(}w \text{)} : \text{Valence of English word (or English phrase) w is SO of word (or phrase) by using the Hamann Coefficient (HC)} \)
6) \( \text{Positive}_\text{query} \): \( \text{HC} \)ive or good or positive or beautiful or strong or nice or excellent or fortunate or correct or superior) with the positive query is the a group of the positive English words.
7) \( \text{Negative}_\text{query} \): (passive or bad or negative or ugly or weak or nasty or poor or unfortunate or wrong or inferior) with the negative query is the a group of the negative English words.
8) \( F(w, \text{positive}_\text{query}) \): number of returned results in Google Search by keyword \( \text{positive}_\text{query} \) and \( w \). We use the Google Search API to get the number of returned results in search online Google by keyword \( \text{positive}_\text{query} \) and \( w \).
9) \( F(w, \text{negative}_\text{query}) \): number of returned results in Google Search by keyword \( \text{negative}_\text{query} \) and \( w \). We use the Google Search API to get the number of returned results in search online Google by keyword \( \text{negative}_\text{query} \) and \( w \).
10) \( F(w) \): number of returned results in Google Search by keyword \( w \). We use the Google Search API to get the number of returned results in search online Google by keyword \( w \).
11) \( P(w, \text{positive}_\text{query}) \): number of returned results in Google Search by keyword \( \text{positive}_\text{query} \) and \( w \). We use the Google Search API to get the number of returned results in search online Google by keyword \( \text{positive}_\text{query} \) and \( w \).
12) \( P(w, \text{positive}_\text{query}) \): number of returned results in the Google Search by keyword \( w \) and \( \text{positive}_\text{query} \). We use the Google Search API to get the number of returned results in search online Google by keyword \( w \) and \( \text{positive}_\text{query} \).
13) \( P(w, \text{negative}_\text{query}) \): number of returned results in the Google Search by keyword \( w \) and \( \text{negative}_\text{query} \). We use the Google Search API to get the number of returned results in search online Google by keyword \( w \) and \( \text{negative}_\text{query} \).
14) \( P(w, \text{positive}_\text{query}) \): number of returned results in Google Search by keyword \( \text{positive}_\text{query} \) and \( w \) and \( \text{negative}_\text{query} \). We use the Google Search API to get the number of returned results in search online Google by keyword \( \text{positive}_\text{query} \) and \( w \) and \( \text{negative}_\text{query} \).
15) \( P(w, \text{negative}_\text{query}) \): number of returned results in the Google Search by keyword \( \text{positive}_\text{query} \) and \( w \) and \( \text{negative}_\text{query} \). We use the Google Search API to get the number of returned results in search online Google by keyword \( \text{positive}_\text{query} \) and \( w \) and \( \text{negative}_\text{query} \).
16) \( P(w, \text{negative}_\text{query}) \): number of returned results in the Google Search by keyword \( \text{negative}_\text{query} \) and \( w \) and \( \text{negative}_\text{query} \). We use the Google Search API to get the number of returned results in search online Google by keyword \( \text{negative}_\text{query} \) and \( w \) and \( \text{negative}_\text{query} \).

According to Cosine, Outli, 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 \( \text{positive}_\text{query} \) with \( w \) and the remote of \( \text{positive}_\text{query} \) with \( w \) and the proximity of \( \text{negative}_\text{query} \) with \( w \) and the remote of \( \text{negative}_\text{query} \) with \( w \).

\( \text{IFHC}(w, \text{positive}_\text{query}) \) is as equal as \( 1 \) the English word \( w \) is the nearest of \( \text{positive}_\text{query} \)
\( \text{IFHC}(w, \text{positive}_\text{query}) \) is as equal as \( 0 \) the English word \( w \) is the farthest of \( \text{positive}_\text{query} \)
\( \text{IFHC}(w, \text{positive}_\text{query}) > 0 \) and \( \text{HC}(w, \text{positive}_\text{query}) = 1, \) the English word \( w \) belongs to \( \text{positive}_\text{query} \) being the group of the English words.
\( \text{IFHC}(w, \text{negative}_\text{query}) \) is as equal as \( 1 \) the English word \( w \) is the nearest of \( \text{negative}_\text{query} \)
\( \text{IFHC}(w, \text{negative}_\text{query}) \) is as equal as \( 0 \) the English word \( w \) is the farthest of \( \text{negative}_\text{query} \)
\( \text{IFHC}(w, \text{negative}_\text{query}) > 0 \) and \( \text{HC}(w, \text{negative}_\text{query}) = 1, \) the English word \( w \) belongs to \( \text{negative}_\text{query} \) being the group of the English words.

So, the valence of the English word \( w \) is the value of \( \text{HC}(w, \text{positive}_\text{query}) \) subtracting the value of \( \text{HC}(w, \text{negative}_\text{query}) \) and the Eq. 8 is the equation of identifying the valence of the English word \( w \). We have the information about \( \text{HC} \) as follows:

1) \( \text{HC}(w, \text{positive}_\text{query}) = 0 \) and \( \text{HC}(w, \text{positive}_\text{query}) = 1 \)
2) \( \text{HC}(w, \text{negative}_\text{query}) = 0 \) and \( \text{HC}(w, \text{negative}_\text{query}) = 1 \)
3) \( \text{HC}(w, \text{positive}_\text{query}) = 0 \) and \( \text{HC}(w, \text{negative}_\text{query}) = 0 \) then \( \text{SOHC}(w) = 0 \)
4) \( \text{HC}(w, \text{positive}_\text{query}) = 1 \) and \( \text{HC}(w, \text{negative}_\text{query}) = 0 \) then \( \text{SOHC}(w) = 0 \)
5) \( \text{HC}(w, \text{positive}_\text{query}) = 0 \) and \( \text{HC}(w, \text{negative}_\text{query}) = 1 \) then \( \text{SOHC}(w) = 0 \)
6) \( \text{HC}(w, \text{positive}_\text{query}) = 0 \) and \( \text{HC}(w, \text{negative}_\text{query}) = 1 \) then \( \text{SOHC}(w) = 0 \)
7) \( \text{SOHC}(w) = 0 \) and \( \text{SOHC}(w) = 0 \)
If $SO_{HC}(w) > 0$, the polarity of the English word $w$ is positive polarity. If $SO_{HC}(w) = 0$, the polarity of the English word $w$ is neutral polarity. In addition, the semantic value of the English word $w$ is $SO_{HC}(w)$.

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. 8-10, we identify the valence and the polarity of one word (or one phrase) in English by using the HC through the Google Search engine with AND operator and OR operator.

We show the comparisons of advantages of the results of our new model with the researches in Table as follows (Table 1-4).


**Table 1:** The sentiment lexicons of the bESD

<table>
<thead>
<tr>
<th>Ordering number</th>
<th>Lexicons</th>
<th>Valence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Good</td>
<td>+1</td>
</tr>
<tr>
<td>2</td>
<td>Very good</td>
<td>+2</td>
</tr>
<tr>
<td>3</td>
<td>Bad</td>
<td>-1</td>
</tr>
<tr>
<td>4</td>
<td>Very bad</td>
<td>-2</td>
</tr>
<tr>
<td>5</td>
<td>Terrible</td>
<td>-1.2</td>
</tr>
<tr>
<td>6</td>
<td>Very terrible</td>
<td>-2.3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>55,000</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

A basis English Sentiment Dictionary (bESD) in a sequential environment: In this part, the valences and the polarities of the English words or phrases for our basis English Sentiment Dictionary (bESD) are calculated by using the HC in a sequential system from at least 55,000 English terms including nouns, verbs, adjectives, etc. according to Anonymous (2017 a-k) as shown in Fig. 4.

The algorithm 2 is proposed to perform this study. The algorithm 2 has the main ideas as follows:

**Algorithm 2; Main idea:**

**Input:** 55,000 English words, 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. 8-10 of the calculating a valence of one word (or one phrase) in English in the study (4.1.1), the sentiment score and the polarity of this term are identified. The valence and the polarity are calculated by using the HC through the Google search engine with AND operator and OR operator.

**Step 3:** Add this term into the basis English Sentiment Dictionary (bESD)

**Step 4:** End Repeat-End Step 1

**Step 5:** Return bESD

We store more 55,000 English words (or English phrases) of our basis English Sentiment Dictionary (bESD) in Microsoft SQL Server 2008 R2.

A basis English Sentiment Dictionary (bESD) in a distributed system: In this part, the valences and the polarities of the English words or phrases for our basis English Sentiment Dictionary (bESD) are identified by using the HC in a parallel network environment from at...

<table>
<thead>
<tr>
<th>Studies</th>
<th>PEM</th>
<th>IM</th>
<th>Language</th>
<th>SD</th>
<th>DT</th>
<th>HC</th>
<th>SC</th>
<th>Other measures</th>
<th>Search engines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bai et al. (2014)</td>
<td>Yes</td>
<td>No</td>
<td>English</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No Mention</td>
</tr>
<tr>
<td>Turney and Littman (2002)</td>
<td>Yes</td>
<td>No</td>
<td>English</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Latent Semantic Analysis (LSA)</td>
<td>AltaVista</td>
</tr>
<tr>
<td>Malouf and Mullen (2017)</td>
<td>Yes</td>
<td>No</td>
<td>English</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Baseline; Turney-inspired NB; Cluster-NB; Human</td>
<td>AltaVista</td>
</tr>
<tr>
<td>Schreibler (2010)</td>
<td>Yes</td>
<td>No</td>
<td>English-German</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>SimRank</td>
<td>Google search engine</td>
</tr>
<tr>
<td>Jovanovski et al. (2015)</td>
<td>Yes</td>
<td>No</td>
<td>English</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No Mention</td>
<td>AltaVista search engine</td>
</tr>
<tr>
<td>Htait et al. (2016)</td>
<td>Yes</td>
<td>No</td>
<td>English-Arabic</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No Mention</td>
<td>Google search engine</td>
</tr>
<tr>
<td>Wan (2009)</td>
<td>Yes</td>
<td>No</td>
<td>English-Chinese</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>SVM(CN); SVM(EN); SVM(ENCN1); SVM(ENCN2);</td>
<td>No mention</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TSVM(CN); TSVM(EN); TSVM(ENCN1); SVM(ENCN2); CSparse</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SO Calculation: SVM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Feature selection</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Expectation Cross Entropy</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Information Gain</td>
<td></td>
</tr>
<tr>
<td>Brooke et al. (2009)</td>
<td>Yes</td>
<td>No</td>
<td>English-Spanish</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Google</td>
</tr>
<tr>
<td>Jiang et al. (2015)</td>
<td>Yes</td>
<td>No</td>
<td>Chinese-Tibetan</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>DF, CHL, MI andG</td>
<td>No mention</td>
</tr>
<tr>
<td>Tan and Zhang (2008)</td>
<td>Yes</td>
<td>No</td>
<td>Chinese</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Information Bottleneck Method (IB), LE</td>
<td>AltaVista</td>
</tr>
<tr>
<td>Du et al. (2010)</td>
<td>Yes</td>
<td>No</td>
<td>Chinese</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Information Bottleneck Method (IB), LE</td>
<td>No mention</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SVM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Harmonic-Mean</td>
<td></td>
</tr>
<tr>
<td>Feng et al. (2013)</td>
<td>Yes</td>
<td>Yes</td>
<td>English</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Dice; NGD</td>
<td>Google search engine</td>
</tr>
<tr>
<td>Nguyen and Hagiwara</td>
<td>Yes</td>
<td>Yes</td>
<td>English</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Dice; Overlap</td>
<td>Google</td>
</tr>
<tr>
<td>Shihalgar and Doit (2014)</td>
<td>No</td>
<td>Yes</td>
<td>English</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>A Jaccard Index-Based Clustering Algorithm (JBCA)</td>
<td>No mention</td>
</tr>
<tr>
<td>Ji et al. (2015)</td>
<td>Yes</td>
<td>Yes</td>
<td>Arabic</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Naive Bayes, Two-Step Multinomial Naive Bayes and</td>
<td>Google</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Two-Step Polynomial-Kernel Support Vector Machine</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Naive Bayes (NB), Support Vector Machines (SVM);</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rocchio; Cosine</td>
<td></td>
</tr>
<tr>
<td>Omar et al. (2013)</td>
<td>Yes</td>
<td>No</td>
<td>Arabic</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>A new score-Economic Value (EV), etc.</td>
<td>Chinese search</td>
</tr>
<tr>
<td>Omar et al. (2013)</td>
<td>Yes</td>
<td>No</td>
<td>Chinese</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No mention</td>
</tr>
<tr>
<td>Ren et al. (2014)</td>
<td>No</td>
<td>Yes</td>
<td>Chinese</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Cosine</td>
<td>No mention</td>
</tr>
<tr>
<td>Neter et al. (2012)</td>
<td>No</td>
<td>Yes</td>
<td>English</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Dice; overlap; Cosine</td>
<td>No mention</td>
</tr>
<tr>
<td>Ren et al. (2011)</td>
<td>No</td>
<td>Yes</td>
<td>Chinese</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Dice; overlap; Cosine</td>
<td>No mention</td>
</tr>
<tr>
<td>Phu et al. (2017a-b)</td>
<td>No</td>
<td>No</td>
<td>Vietnamese</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Ochiai measure</td>
<td>Google</td>
</tr>
<tr>
<td>Phu et al. (2017a-b)</td>
<td>No</td>
<td>No</td>
<td>English</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Cosine coefficient</td>
<td>Google</td>
</tr>
<tr>
<td>Phu et al. (2017a-b)</td>
<td>No</td>
<td>No</td>
<td>Vietnamese</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Sorenson measure</td>
<td>Google</td>
</tr>
<tr>
<td>Phu et al. (2017a-b)</td>
<td>No</td>
<td>No</td>
<td>English</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Jaccard</td>
<td>Google</td>
</tr>
<tr>
<td>Our research</td>
<td>No</td>
<td>No</td>
<td>English</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Tanimoto coefficient</td>
<td>Google Search engine</td>
</tr>
</tbody>
</table>

Hamann Coefficient (HC); Semantic classification, Sentiment Classification (SC)

least 55,000 English terms including nouns, verbs, adjectives, etc. based on Anonymous (2017 a-k) as shown in Fig. 5.

This study in Fig. 5 comprises two phases as follows: the Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase is the 55,000 terms in English by Anonymous (2017 a-k). 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).

The algorithm 3 is build to implement the Hadoop Map phase of creating a basis English Sentiment Dictionary (bESD) in a distributed environment. The algorithm 3 has its main ideas as follows:

<table>
<thead>
<tr>
<th>Surveys</th>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bai et al. (2014)</td>
<td>Constructing sentiment lexicons in Norwegian from a large text corpus</td>
<td>Through the researcher’s PMI computations in this survey, they used a distance of 100 words from the seed word, but it might be that other lexicons that generate better sentiment lexicons. Some of the researcher’s preliminary research showed that 100 gave a better result</td>
<td>The researcher’s need to investigate this more closely to find the optimal distance. Another factor that has not been investigated much in the literature is the selection of seed words. Since they are the basis for PMI calculation, it might be a lot to gain by finding better seed words. The researcher’s 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>Turney and Littman (2002)</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-ISA. The accuracy of SO-PMI-IR is comparable to the accuracy of HBM, the algorithm of Hatzivasiloglou and McKeown. SO-PMI-IR requires a large corpus but it is simple, easy to implement, unsupervised and it is not restricted to adjectives.</td>
<td>No mention</td>
</tr>
<tr>
<td>Malouf and Mullen (2017)</td>
<td>Graph-based user classification for informal online political discourse</td>
<td>The researcher’s describe several experiments identifying the political orientation of posters in an informal environment. The researcher’s results indicate that the most promising approach is to augment text classification methods by exploiting information about how posters interact with each other.</td>
<td>There is still much left to investigate in terms of optimizing the linguistic analysis, beginning with spelling correction and working up to shallow parsing and co-reference identification. Likewise, it will also be worth while to further investigate exploiting sentiment values of phrases and clauses, taking cues from methods.</td>
</tr>
<tr>
<td>Scheible (2010)</td>
<td>A novel graph-based approach using SimRank</td>
<td>The researcher’s presented a novel approach to the translation of sentiment information that outperforms SOPMI, an established method. In particular, the researcher’s could show that SimRank outperforms SO-PMI for values of the threshold x in an interval that most likely leads to the correct separation of positive, neutral and negative adjectives.</td>
<td>The researcher’s future research will include a further examination of the merits of its application for knowledge-sparse languages.</td>
</tr>
<tr>
<td>Jovanoski et al. (2015)</td>
<td>Analysis in Twitter for Macedonian</td>
<td>The researcher’s experimental results show an F1-score of 0.16 which is very strong and is on par with the best results for English which were achieved in recent SemEval competitions.</td>
<td>In future research, the researcher’s are interested in studying the impact of the raw corpus size, e.g., the researcher’s could only collect half a million Tweets for creating lexicon and analyzing evaluating the system while Krichen et al. built their lexicon on million Tweets and evaluated their system on 1.5 million English Tweets. Moreover, the researcher’s 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>H haircut et al. (2016)</td>
<td>Using web search engines for English and Arabic unsupervised sentiment intensity prediction</td>
<td>For the general English sub-task, the researcher’s system has modest but interesting results. For the mixed polarity English sub-task, the researcher’s system results achieve the second place. For the Arabic phrases sub-task, the researcher’s system has very interesting results since they applied the unsupervised method only.</td>
<td>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>Wan (2009)</td>
<td>Co-training for cross-lingual sentiment classification</td>
<td>The researcher’s 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.</td>
<td>In future research, the researcher’s will improve the sentiment classification accuracy in the following two ways: the smoothed co-training approach used in Mihalcea will be adapted for employ sentiment classification. The researcher’s will the Structural Correspondence Learning domain adaption algorithm used by Sitzer (SCL) et al. for linking the translated text and the natural text.</td>
</tr>
<tr>
<td>Brooke et al. (2009)</td>
<td>Cross-linguistic sentiment analysis: From English to Spanish</td>
<td>Our Spanish SO calculator (SOCAL) is clearly inferior to the researcher’s 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>
<td>No mention</td>
</tr>
<tr>
<td>Surveys</td>
<td>Approach</td>
<td>Advantages</td>
<td>Disadvantages</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Jiang et al.</td>
<td>Micro-blog emotion orientation analysis algorithm based on Tibetan and Chinese mixed text</td>
<td>By emotion orientation analyzing and studying of Tibetan microblog which is concerned in Sina, making Tibetan Chinese emotion dictionary, Chinese sentences, Tibetan part of speech sequence and emotion symbol as emotion factors and using expected cross entropy combined fuzzy set to do feature selection to realize a kind of microblog emotion orientation analyzing algorithm based on Tibetan and Chinese mixed text. The experimental results showed that the method can obtain better performance in Tibetan and Chinese mixed Microblog orientation analysis.</td>
<td>No mention</td>
</tr>
<tr>
<td>Tan and Zhang</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, Naive Bayes and SVM) are investigated on a Chinese sentiment corpus with a size of 1201 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 researcher's found that sentiment classifiers are severely dependent on domains or topics.</td>
<td>No mention</td>
</tr>
<tr>
<td>Du et al.</td>
<td>Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon</td>
<td>The researcher's theory verifies the convergence property of the proposed method. The empirical results also support the researcher's theoretical analysis. In their experiment, it is shown that the proposed method greatly outperforms the baseline methods in the task of building out-of-domain sentiment lexicon.</td>
<td>In this study, only the mutual information measure is employed to measure the three kinds of relationship. In order to show the robustness of the framework, the researcher's future effort is to investigate how to integrate more measures into this framework.</td>
</tr>
<tr>
<td>Zhang et al.</td>
<td>Sentiment classification for consumer word-of-mouth in Chinese: comparison between supervised and unsupervised approaches</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 number of training examples is smaller than 100.</td>
<td>No mention</td>
</tr>
<tr>
<td>Wang and Araki</td>
<td>Modifying SO-PMI for Japanese weblogs opinion mining by using a balancing factor and detecting neutral expressions</td>
<td>After these modifications, the researcher's achieved a well-balanced result: both positive and negative accuracy exceeded 79%. This shows that the researcher's proposed approach not only adapted the SO-PMI for Japanese but also modified it to analyze Japanese opinions more effectively.</td>
<td>In the future, the researcher's will evaluate different choices of words for the sets of positive and negative reference words. The researcher's also plan to appraise their proposal on other languages.</td>
</tr>
<tr>
<td>Feng et al.</td>
<td>In this survey, the researcher's empirically evaluated the performance of different corpora in sentiment similarity measurement which is the fundamental task for word polarity classification</td>
<td>Experiment results show that the Twitter data can achieve a much better performance than the Google, WebT1 and Wikipedia based methods.</td>
<td>No mention</td>
</tr>
<tr>
<td>Nguyen and Hagiwara</td>
<td>Adjective-based estimation of short sentence's impression</td>
<td>The adjectives are ranked and top three adjectives are considered as an output of system. For example, the experiments were carried out and got fairly good results. With the input “it is snowy”, the results are white (0.70), light (0.49), cold (0.43), solid (0.38) and scenic (0.37).</td>
<td>In the researcher’s future research, 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>
</tr>
<tr>
<td>Shikilagor and Dect</td>
<td>Jaccard index based clustering algorithm for mining online review</td>
<td>In this research, 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>For future research, by using this framework, it can extend it to predicting sales performance in the other domains like customer electronics, mobile phones, computers based on the user reviews posted on the website, etc.</td>
</tr>
<tr>
<td>Ji et al.</td>
<td>Twitter sentiment classification for measuring public health concerns</td>
<td>Based on the number of tweets classified as personal negative, the researcher’s compute a Measure of Concern (MOC) and a timeline of the MOC. We attempt to correlate peaks of the MOC timeline to the peaks of the News (Non-Personal) timeline. The researcher’s best Accuracy results are Achieved using the two-step method with a Naive Bayes classifier for the Epidemic domain (six datasets) and the mental health domain (three datasets).</td>
<td>No mention</td>
</tr>
<tr>
<td>Surveys</td>
<td>Approach</td>
<td>Advantages</td>
<td>Disadvantages</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Omar et al. (2013)</td>
<td>Ensemble of classification algorithms for subjectivity and sentiment analysis of Arabic customer’s reviews</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>
<td>No mention</td>
</tr>
<tr>
<td>Mao et al. (2014)</td>
<td>Automatic construction of financial semantic orientation lexicon from large-scale Chinese News Corpus</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 researcher’s propose a novel idea to construct a financial semantic orientation lexicon from large-scale Chinese news corpus automatically</td>
<td>No mention</td>
</tr>
<tr>
<td>Ren et al. (2014)</td>
<td>Sentiment classification in under-resourced languages using graph-based semi-supervised learning methods</td>
<td>In particular, the researcher’s found that choosing initially labeled vertices in H.C. order with their degree and PageRank score can improve the performance. However, pruning unreliable edges will make things more difficult to predict. The researcher’s believe that other people who are interested in this field can benefit from their empirical findings.</td>
<td>As future research, first, the researcher’s will attempt to use a sophisticated approach to induce better sentiment features. The researcher’s consider such elaborated features improve the classification performance, especially in the book domain. The researcher’s also plan to exploit a much larger amount of unlabeled data to fully take advantage of SSL algorithms.</td>
</tr>
<tr>
<td>Netzer et al. (2012)</td>
<td>A text-mining approach and combine it with semantic network analysis tools</td>
<td>No mention</td>
<td>No mention</td>
</tr>
<tr>
<td>Ren et al. (2011)</td>
<td>Sentiment classification in resource-scarce languages using label propagation</td>
<td>The researcher’s 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 and 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.</td>
<td>No mention</td>
</tr>
<tr>
<td>Phu et al. (2017a-h)</td>
<td>A Vietnamese adjective emotion dictionary based on exploitation of Vietnamese language characteristics</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, the Vietnamese adjectives bring sentiment and their semantic values (or their sentiment scores) are not changed in any context, thereby the results of the emotion classification are not high accuracy. The researcher’s propose many rules based on Vietnamese language characteristics to determine the emotional values of the Vietnamese adjective phrases bearing sentiment in specific contexts. The researcher’s Vietnamese sentiment adjective dictionary is widely used in applications and researches of the Vietnamese semantic classification.</td>
<td>Not calculating all Vietnamese words completely; not identifying all Vietnamese adjective phrases fully, etc.</td>
</tr>
<tr>
<td>Phu et al. (2017a-h)</td>
<td>A valences-totaling model for English sentiment classification</td>
<td>The researcher’s present a full range of English sentences, thus, the emotion expressed in the English text is classified with more precision. The researcher’s new model is not dependent on a special domain and training data set it is a domain-independent classifier. The researcher’s 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>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>Phu et al. (2017a-h)</td>
<td>Shifting semantic values of English phrases for classification</td>
<td>The results of the sentiment classification are not high accuracy if the English phrases bring the emotions and their semantic values (or their sentiment scores) are not changed in any context. For these reasons, the researcher’s propose many rules based on English language grammars to calculate the sentimental values of the English phrases bearing emotion in their specific contexts.</td>
<td>This survey is only applied to the English adverb phrases. The proposed model is needed to research more and more for the different types of the English words such as English noun, English adverbs, etc.</td>
</tr>
</tbody>
</table>

2066
Table 3: Continue

<table>
<thead>
<tr>
<th>Surveys</th>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phu et al. (2017a-k)</td>
<td>A valence-totaling model for Vietnamese sentiment classification</td>
<td>The results of this research are widely used in applications and researches of the English semantic classification. The researcher’s have used the VTMV to classify 30,000 Vietnamese documents which include the 15,000 positive Vietnamese documents and the 15,000 negative Vietnamese documents. The researcher’s have achieved accuracy in 63.9% of the researcher’s Vietnamese testing data set. VTMV is not dependent on the special domain. VTMV is also not dependent on the training data set and there is no training stage in this VTMV. From the researcher’s results in this research, our VTMV can be applied in the different fields of the Vietnamese natural language processing. In addition, the researcher’s TCMV 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.</td>
<td>It has a low accuracy</td>
</tr>
</tbody>
</table>

Phu et al. (2017a-k) | Semantic lexicons of English nouns for classification | The proposed rules based on English language grammars to calculate the sentimental values of the English phrases bearing emotion in their specific contexts. The results of the sentiment classification are not high accuracy if the English phrases bring the emotions and their semantic values (or their sentiment scores) are not changed in any context. The valences of the English words (or the English phrases) are identified by using Tanimoto Coefficient (TC) through the Google search engine with AND operator and OR operator. The emotional values of the English noun phrases are based on the English grammars (English language characteristics). | This survey is only applied in the English noun phrases. The proposed model is needed to research more and more about the different types of the English words such as English English adverbs, etc. |

Our research: Hannan Coefficient (HC) through the Google search engine with AND operator and OR operator; We use the Hannan Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Function (FF) which is the Roulette-Wheel Selection (RWS) 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.

![Diagram](image)

**Fig. 4:** Overview of creating a basis English Sentiment Dictionary (bESD) in a sequential environment
Table 4: Comparisons of our model’s results with the researches related to the Hamann Coefficient (HC) by Choi et al. (2010), Schraw (1995), David et al. (1959), Schraw et al. (2004), Griggsby et al. (1998) and Urbina et al. (2010)

<table>
<thead>
<tr>
<th>Studies</th>
<th>PMI</th>
<th>JM</th>
<th>Hamann Coefficient (HC)</th>
<th>Language</th>
<th>SD</th>
<th>DT</th>
<th>Sentiment classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choi et al. (2010)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>English</td>
<td>NM</td>
<td>NM</td>
<td>No mention</td>
</tr>
<tr>
<td>Schraw (1995)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>No mention</td>
</tr>
<tr>
<td>David et al. (1959)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>No mention</td>
</tr>
<tr>
<td>Schraw et al. (2004)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>No mention</td>
</tr>
<tr>
<td>Griggsby et al. (1998)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>No mention</td>
</tr>
<tr>
<td>Urbina et al. (2010)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>No mention</td>
</tr>
<tr>
<td>Our research</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>English Language</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 5: Comparisons of our model’s benefits and drawbacks with the studies related to the Hamann Coefficient (HC) by Choi et al. (2010), Schraw (1995), David et al. (1959), Schraw et al. (2004), Griggsby et al. (1998) and Urbina et al. (2010)

<table>
<thead>
<tr>
<th>Surveys</th>
<th>Approach</th>
<th>Benefits</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choi et al. (2010)</td>
<td>A survey of binary similarity and distance measures</td>
<td>Applying appropriate measures results in more accurate data analysis</td>
<td>No mention</td>
</tr>
<tr>
<td>Schraw (1995)</td>
<td>Measures of feeling-of-knowing accuracy: A new look at an old problem</td>
<td>A proof is provided which reveals a lack of one-to-one relation between the two and suggests that they may be independent under most circumstances. It is concluded that both measures should be reported together as complementary indices in each captures a different facet of feeling-of-knowing performance. Alternative measures for gamma and the Hamann coefficient are considered and a number of recommendations are made for future research.</td>
<td>No mention</td>
</tr>
<tr>
<td>David et al. (1959)</td>
<td>The second virial coefficients of some cyclic hydrocarbons</td>
<td>This survey reports some measurements of the second virial coefficient B of cyclopropane in the temperature range 300-400K. It also gives some values of B for cyclohexane and benzene, derived from critical analyses of the published vapour densities and P-V-T properties of these gases between 300 and 650K</td>
<td>No mention</td>
</tr>
<tr>
<td>Schraw et al. (2004)</td>
<td>A role for right medial prefrontal cortex in accurate feeling-of-knowing judgments: evidence from patients with lesions to frontal cortex</td>
<td>While frontal patients were impaired at recall and recognition memory, they were able to make accurate confidence judgments about their recall attempts. By contrast, as a group, the patients were markedly impaired in the accuracy of their prospective FOK judgments. Lesion analysis of frontal patients with clear FOK impairment revealed an overlapping region of damage in right medial prefrontal cortex. These findings provide functional and anatomical evidence for a dissociation between recall confidence and prospective memory monitoring and are discussed in terms of familiarity and access theories of FOK predictions.</td>
<td>No mention</td>
</tr>
<tr>
<td>Griggsby et al. (1998)</td>
<td>Executive cognitive abilities and functional status among community-dwelling older persons in the san diego valley health and aging study</td>
<td>Both general mental status and executive functioning demonstrated statistically significant univariate associations with all seven functional status measures (both self-report and observed performance). In a series of ordinary least squares regression models, executive functioning was a predictor for self-reported ADLs and observed performance of complex IADL tasks such as managing money and medications. Mental status did not predict self-reported functioning but was a predictor of observed performance. Depression was a significant variable for self-report measures but not for observed performance. Executive functioning and general mental status demonstrated some degree of independence from one another.</td>
<td>No mention</td>
</tr>
<tr>
<td>Urbina et al. (2010)</td>
<td>Prevalence of increased arterial stiffness in children with type 1 diabetes mellitus differs by measurement site and sex: the search for diabetes in youth study</td>
<td>Subjects with T1DM had higher body mass index, LDL-cholesterol, fasting glucose and blood pressure than control subjects. Diabetic subjects had lower BrachID and higher A1c&gt;75 indicating increased stiffness. Age-adjusted pulse wave velocity (aortic-femoral) was higher in cases (all p&lt;0.05). Increased peripheral stiffness was more common than central stiffness in subjects with T1DM (low BrachID in 33% vs high PWV in 9.9%). Male subjects with type 1 diabetes had a higher prevalence of VS abnormalities than females. Presence of T1DM, male sex and increased mean arterial pressure were the most consistent independent determinants of vascular stiffness.</td>
<td>No mention</td>
</tr>
</tbody>
</table>

Our research: Hamann Coefficient (HC) through the Google search engine with AND operator and OR operator. We use the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Function (FF) which is the Roulette Wheel Selection (RWS) 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 study.

Algorithm 3: 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
1. Step 1: Each term in the 55,000 terms, do repeat:
   a. Step 2: By using Eq. 8.10 of the calculating valence of one word (or one phrase) in English in the study, the sentiment score and the polarity of this term are identified. The valence and the polarity are calculated by using the HC through the Google search engine with AND operator and OR operator.
   b. Step 3: Return this term
The algorithm 4 is proposed to perform the Hadoop Reduce phase of creating a basis English Sentiment Dictionary (bESD) in a distributed environment. The algorithm 4 comprises the main ideas as follows:

**Algorithm 4: 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: Receive this term
- Step 2: Add this term into the basis English Sentiment Dictionary (bESD)
- Step 3: Return bESD

At least 55,000 English words (or English phrases) of our basis English Sentiment Dictionary (bESD) are stored in Microsoft SQL Server 2008 R2

Implementing the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) in both a sequential environment and a distributed network system: Figure 6 shows that this compiles two parts as follows: in the first part, we use the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) to classify the documents of the testing data set into either the positive vector group or the negative vector group in a sequential environment. In the second sub-section, we use the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) to cluster the documents of the testing data set into either the positive vector group or the negative vector group in a distributed system.

We encrypt the sentiment lexicons of the bESD to the bit arrays and each bit array in the bit arrays presents each term in the sentiment lexicons with the information as follows: a content of this term, a sentiment score of this term. This is called the bit arrays of the bESD which are stored in a small storage space. We assume that the sentiment lexicons of the bESD is stored in Table 5.

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

Based on the English Dictionaries (Anonymous, 2017a-k), the longest word in English has 189,819 letters. According to the binary code of letters in English by Anonymous (2017a-k), we see that the 7 binary bits are used in encode one letter in all the letters in English. Therefore, we need 189,819 (letters) * 7 (bits) = 1,328,733 (bits) to present one word in English.

So, we need (1,328,733 bits of the content + 8 bits of the valence) = 1,328,741 bits to show fully one sentiment lexicon of the bESD in Fig. 7.

We transfer the valence of one sentiment lexicon of the bESD to a binary code based on the transferring a decimal to a binary code by Anonymous (2017a-k). We present the information about the GA briefly as follows: according to Davis (1991), Kora and Krishna (2016), Yang et al. (2016), Erkaya and Uzmay (2016) and Wu et al. (2016), we show that the basic operations of the Genetic algorithm, at the same time, also used for the GA in our sequential environment and our parallel network environment. The Genetic algorithm (GA: Genetic Algorithms) and other evolutionary algorithms based on forming the notion that natural evolutionary process is reasonable, perfect. It stems from the evofled idea to survive and grow in the wild. GA is a problem-solving method to mimic the behavior of humans in order to survive and develop. It helps to find the optimal solution.
and the best in terms of time and space allow. GA considers all solutions by at least some solutions then eliminates the irrelevant components and select the relevant components more adapted to create birth and evolution aimed at creating solutions which have a new adaptive coefficient increasing. The adaptive coefficient is used as a gauge of the solution. The main steps of the GA algorithm 5:

**Algorithm 5: Genetic algorithm:**

Step 1: Select models to symbolize the solutions. The models can be sequence (string) of the binary number: 1 and 0, decimal and can be letters or mixture letters and numbers

Step 2: Select the adaptive function (or the Fitness function) to use as a gauge of the solution

Step 3: Continue the transformation form until achieving the best solution or until the termination of the time

**Genetic operators and genetic operations**

**Reproductive operator:** Reproductive operator includes two processes: the reproduction process (allowing regeneration), the selection process (selection).

**Allowing regeneration:** Allowing regeneration is the process which allows chromosomes to copy on the basis of the adaptive coefficient. The adaptive coefficient is a function which is assigned the real value, corresponding to each chromosome in the population. This process is described as follows: determine the adaptive coefficient of each chromosome in the population at generation t, tabulate cumulative adaptive values (in order assigned to each chromosome).

Suppose, the population has n individuals. Call the adaptive coefficient of the corresponding chromosome is $f_i$, cumulative total is $f_t$ which is defined by:

$$f_t = \sum_{j=1}^{n} f_j$$

Call $F_n$ is the sum of the adaptive coefficient in all the population. Pick a random number $f$ between 0 and $F_n$. Select the first instance correspond $f - f_t$ into new population.
**Selection process (selection):** The selection process is the process of removing the poor adaptive chromosomes in the population. This process is described as follows:

- Arrange population in order of descending degree of adaptation
- Remove the chromosome in the last of the sequence. Keep n in the best individuals

**Crossover operator (crossover):** Crossover is the process of creating the new chromosomes based on the father-mother chromosomes by grafting a segment on the father-mother chromosomes together. The crossover operator is assigned with a probability pc. This process is described as follows: randomly select a pair of chromosomes (father-mother) in the population. Suppose, the father-mother chromosomes have the same length m.

Create a random number in the range from 1 to m-1 (called as cross coupling point). The cross coupling point divides the father-mother chromosomes into two sub-strings which have lengths m1, m2. The two new sub-strings created, is: m11+m22 and m21+m12. Put the two new chromosomes into the population. Example:

![Crossing Over Example](image)

**Mutation operator (mutation):** Mutation is a phenomenon which the child chromosomes carry some features not in the genetic code of the father-mother chromosomes.

- Choose a random chromosome in the population
- Create a random number k between 1 and m, 1 ≤ k ≤ m
- Change bit k. Put this chromosome in the population to participate in the evolution of the next generation

Example: 0100111

Example of the sexual reproduction intersecting 3 bits:

**Father:** 10011110 —— Child 1: 11111110
**Mother:** 11111000 —— Child 2: 10011000

**Fitness** is defined as an objective function that quantifies the optimality of a solution (chromosome) to the target problem. How to choose fitness is dependent on the problem that we study. Choosing the different fitness function will give different results. In this survey, we use the Roulette-Wheel Selection (RWS).

According to the researches related to the Roulette-Wheel Selection (RWS) by Panda et al. (2009), Lee et al. (1998), and Tao (2003), Lipowski and Lipowska (2012), Zou et al. (2006), Roulette Wheel is the simplest selection approach. In this method, all the chromosomes (individuals) in the population are placed on the roulette wheel according to their fitness value. Each individual is assigned a segment of the roulette wheel. The size of each segment in the roulette wheel is proportional to the value of the fitness of the individual the bigger the value is the larger the segment is. Then, the virtual roulette wheel is spun. The individual corresponding to the segment on which roulette wheel stops are then selected. The process is repeated until the desired number of individuals is selected. Individuals with higher fitness have more probability of selection. This may lead to biased selection towards high fitness individuals. It can also possibly miss the best individuals of a population. There is no guarantee that good individuals will find their way into the next generation. Roulette wheel selection uses an exploitation technique in its approach.

**Performing the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the fitness function (FF) in a sequential environment:** In this study, the Genetic Algorithm (GA) is used with the Roulette-Wheel Selection (RWS) the Fitness Function (FF) to cluster the documents of the testing data set into either the positive polarity or the negative polarity in the sequential environment.

Firstly, we build the sentiment lexicons of the bESD based on a basis English Sentiment Dictionary (bESD) in a sequential environment. We build the algorithm 4 to encrypt one sentiment lexicon (comprising the content and the valence) to a binary array in the sequential environment. The main ideas of the algorithm 5 are as follows:

Example of asexual reproduction:

**Father:** 01101110 —— Child 1: 01110110
**Mother:** 11001110 —— Child 2: 11001110

Example of sexual reproduction intersecting half:

**Father:** 10001110 —— Child 1: 00011100
**Mother:** 01110011 —— Child 2: 11100111

Each pair of parents bears two children in one of the following two methods:

- **Asexual reproduction:** Each child is an exact copy of each father or each mother. Example:
  
  **Father:** 01101100 —— Child 1: 01101100
  **Mother:** 11001110 —— Child 2: 11001110

- **Sexual reproduction (crossover):** Some bits are copied from the mother or a few bits are copied from the father. Example of the sexual reproduction intersecting half:
Fig. 7: Overview of presenting one sentiment lexicon of the bESD in a binary code

**Algorithm 5; Encrypt one sentiment lexicon:**
Input: one sentiment lexicon of the bESD  
Output: a bit array  
Step 1: Split this term into the letters  
Step 2: Set ABitArray := null  
Step 3: Set Valence := Get a valence of this term based on the bESD  
Step 4: Each letter in the letters do repeat  
Step 5: Based on the binary code of letters in English by Anonymous (2017 a), we get a bit array of this letter  
Step 6: Add the bit array of this letter into ABitArray  
Step 7: End Repeat  
End Step 3  
Step 8: Based on the transferring a decimal to a binary code by Anonymous (2017 a), we transfer the valence to a bit array  
Step 9: Add this bit array into ABitArray  
Step 10: Return ABitArray

We propose the algorithm 5 to encode one sentence in English to a binary array in the sequential system. The main ideas of the algorithm 6 are as follows:

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

We encrypt all the positive sentences of the training data set to the positive bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the algorithm 6, called the positive bit array group. The main ideas of the algorithm 7 are as follows (Fig. 7):

**Algorithm 7; Positive sentences of the training data set:**
Input: all the positive sentences of the training data set  
Output: a positive bit array group  
Step 1: Set A PositiveBitArray Group := null  
Step 2: Each sentence in the positive sentences, do repeat  
Step 3: ABitArray = the algorithm 5 to encode one sentence in English to a binary array in the sequential system with the input is this sentence  
Step 4: Add ABitArray into APositiveBitArray Group  
Step 5: End Repeat-End Step 2  
Step 6: Return APositiveBitArray Group

We encode all the negative sentences of the training data set to the negative bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the algorithm 7, called the negative bit array group. The main ideas of the algorithm 8 are as follows:

**Algorithm 8; Negative sentences of the training data set:**
Input: all the negative sentences of the training data set  
Output: a negative bit array group  
Step 1: Set A NegativeBitArray Group := null  
Step 2: Each sentence in the positive sentences, do repeat  
Step 3: ABitArray = the algorithm 5 to encode one sentence in English to a binary array in the sequential system with the input is this sentence  
Step 4: Add ABitArray into ANegativeBitArray Group  
Step 5: End Repeat-End Step 2  
Step 6: Return ANegativeBitArray Group

We propose the algorithm 9 to transfer one document of the testing data set into the bit arrays of the document in the sequential system. The main ideas of the algorithm 8 are as follows (Fig. 8):

**Algorithm 9; Transfer one document of the testing data set:**
Input: one document of the testing data set  
Output: the bit arrays of the document  
Step 1: Set The BitArrays Of The Document := null  
Step 2: Split this document into the sentences  
Step 3: Each sentence in the sentences do repeat  
Step 4: ABitArray = the algorithm 5 to encode one sentence in English to a binary array in the sequential system with the input is this sentence  
Step 5: Add ABitArray into The BitArrays Of The Document  
Step 6: End Repeat-End Step 3  
Step 7: Return The BitArrays Of The Document

From Fig. 8, we show the diagram of the GA in the sequential environment as follows. We build the algorithm 10 to cluster one bit array (corresponding to one
sentence) of the document into either the positive bit array group or the negative bit array group of the training data set by using the GA with the RWS in the sequential system. The main ideas of the algorithm 10 are as follows:

**Algorithm 10; Cluster one bit array:**
Input: one bit array (corresponding to one sentence) of the document the positive bit array group and the negative bit array group the training data set
Output: the sentiments (positive, negative or neutral)
Step 1: randomly initialize population (t)
Step 2: determine fitness of population (t)
Step 3: repeat
Step 4: select parents from population (t)
Step 5: perform crossover on parents creating population(t+1)
Step 6: perform mutation of population (t+1)
Step 7: determine fitness of population (t+1)
Step 8: until best individual is good enough
Step 9: Return this bit array clustered into either the positive bit array group or the negative bit array group of the training data set

We propose the algorithm 11 to cluster one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the sequential environment. The main ideas of the algorithm 11 are as follows:

**Algorithm 11; Cluster one document of the testing data set:**
Input: one document of the testing data set; the positive bit array group and the negative bit array group of the training data set
Output: the sentiments (positive, negative or neutral)
Step 1: ABitArrayGroupOfOneDocument = the algorithm 8 to transfer one document of the testing data set into the bit arrays of the document in the sequential system with the input is this document
Step 2: Set count_positive = 0 and count_negative = 0
Step 3: Each bit array in A BitArray Group of One Document, do repeat
Step 4: OneResult = the algorithm 9 to cluster one bit array (corresponding to one sentence) of the document into either the positive bit array group or the negative bit array group of the training data set by using the GA with the RWS in the sequential system with the input is this bit array, the positive bit array group and the negative bit array group of the training data set
Step 5: If One Result is the positive Then count_positive = count_positive + 1
Step 6: Else If One Result is the negative Then count_negative = count_negative + 1
Step 7: End Repeat - End Step 3
Step 8: If count_positive is greater than count_negative Then Return positive
Step 9: Else If count_positive is less than count_negative Then Return negative
Step 10: Return neutral

We build the algorithm 12 to cluster the documents of the testing data set into either the positive or the negative in the sequential environment. The main ideas of the algorithm 11 area as follows:

**Algorithm 12: The positive or the negative in the sequential environment**
Input: the testing data set and the training data set
Output: the results of the sentiment classification of the testing data set
Step 1: The valences and the polarities of the sentiment lexicons of the bESD are calculated based on a basis English Sentiment Dictionary (bESD) in a sequential environment (4.1.2)
Step 2: A positive bit array group: = encode all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the algorithm 6 with the input is the positive sentences of the training data set
Step 3: A negative bit array group: = encode all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the algorithm 7 with the input is the positive sentences of the training data set
Step 4: Set The Results of the Testing Data Set: = null
Step 5: Each document in the documents of the testing data set, do repeat
Step 6: One Result = the algorithm 11 to cluster one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the sequential environment with the input is this document, the positive bit array group and the negative bit array group
Step 7: Add OneResult into the Results of the testing Data Set
Step 8: End Repeat - End Step 5
Step 9: Return The Results of the Testing Data Set

**Implementing the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) in both a distributed network system:** In this study, the Genetic Algorithm (GA) is used with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) to cluster the documents of the testing data set into either the positive polarity or the negative polarity in the Cloudera parallel network environment.

Firstly, we build the sentiment lexicons of the bESD based on a basis English Sentiment Dictionary (bESD) in a distributed system.

From Fig. 9, we build the algorithm 13 and the algorithm 13 to encrypt one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment. This stage in Fig. 9 comprises two phases as follows: Hadoop Map phase and Hadoop Reduce phase. The input of the Hadoop Map is one sentiment lexicon of the bESD. The output of the Hadoop Map is a bit array of one letter. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is a bit array of one letter. The output of the Hadoop Reduce is a bit array of the term.

We propose the algorithm 12 to implement the Hadoop Map phase of encrypting one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment. The main ideas of the algorithm 13 are as follows:
Fig. 9: Overview of encrypting one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment.

**Algorithm 13: Implement the Hadoop Map phase of encrypting**
Input: one sentiment lexicon of the bESD
Output: a bit array of one letter
Step 1: Input this term and the bESD into the Hadoop Map in the Cloudra system
Step 2: Split this term into the letters
Step 3: Set valence: Get a valence of this term based on the bESD
Step 4: Each letter in the letters, do repeat
Step 5: Based on the binary code of letters in English by Anonymous (2017-g-f), we get a bit array of this letter
Step 6: Return the bit array of this letter; //the output of the Hadoop Map

We build the algorithm 14 to perform the Hadoop Reduce phase of encrypting one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment. The main ideas of the algorithm 14 are as follows:

**Algorithm 14: Perform the Hadoop Reduce phase of encrypting one sentiment lexicon**
Input: the bit array of this letter; //the output of the Hadoop Map
Output: a bit array of the term - ABitArray
Step 1: Receive the bit array of this letter
Step 2: Add the bit array of this letter into ABitArray
Step 3: If this term is full Then
Step 4: Based on on the transferring a decimal to a binary code in Anonymous (2017-m-f), we transfer the valence to a bit array
Step 5: Add this bit array into ABitArray
Step 6: End IF End Step 3
Step 7: Return ABitArray

From Fig. 10, we build the algorithm 15 and the algorithm 16 to encode one sentence in English to a binary array in the distributed system. This stage in Fig. 10 comprises two phases as follows: Hadoop Map phase and Hadoop Reduce phase. The input of the Hadoop Map is one sentence. The output of the Hadoop Map is a bit array of one term -ABitArray. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is a bit array of one term ABitArray. The output of the Hadoop Reduce is a bit array of the sentence ABitArray of Sentence.

We propose the algorithm 15 to perform the Hadoop Map phase of encoding one sentence in English to a binary array in the parallel system. The main ideas of the algorithm 15 are as follows:

**Algorithm 15: Perform the Hadoop Map phase of encoding one sentence in English**
Input: one sentence
Output: a bit array of one term ABitArray
Step 1: Input this sentence into the Hadoop Map in the Cloudera system
Step 2: Split this sentence into the meaningful terms (meaningful word or meaningful phrase)
Step 3: Each term in the terms, do repeat
Step 4: ABitArray = the encrypting one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment in Fig. 9 with the input is this term
Step 5: Return ABitArray

We propose the algorithm 16 to implement the Hadoop Reduce of encoding one sentence in English to a binary array in the parallel system. The main ideas of the algorithm 16 are as follows:

**Algorithm 16: Implement the Hadoop Reduce of encoding one sentence in English:**

Input: a bit array of one term ABitArray the output of the Hadoop Map
Output: a bit array of the sentence A_bitArray of sentence
Step 1: Receive ABitArray
Step 2: Add ABitArray into ABitArray of Sentence
Step 3: Return ABitArray of Sentence

From Fig. 11, we build the algorithm 17 and the algorithm 18 to encrypt all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the positive bit array group. This stage in Fig. 11 comprises two phases as follows: Hadoop Map phase and Hadoop Reduce phase. The input of the Hadoop Map is all the positive sentences of the training data set. The output of the Hadoop Map is ABitArray a bit array of one sentence. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is ABitArray a bit array of one sentence. The output of the Hadoop Reduce is a positive bit array group APositiveBitArrayGroup.

We encrypt all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the algorithm 17 to perform the Hadoop Map phase of this stage in the distributed environment, called the positive bit array group. The main ideas of the algorithm 17 are as follows:

**Algorithm 17: Perform the Hadoop Map phase of this stage in the distributed environment:**

Input: all the positive sentences of the training data set
Output: ABitArray a bit array of one sentence
Step 1: Input all the positive sentences of the training data set into the Hadoop Map in the Cloudera system
Step 2: Each sentence in the positive sentences, do repeat
Step 3: ABitArray = the encoding one sentence in English to a binary array in the parallel system in Fig. 10 with the input is this sentence
Step 4: Return ABitArray

We encrypt all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the algorithm 18 perform the Hadoop Reduce phase of this stage in the parallel system, called the positive bit array group. The main ideas of the algorithm 18 are as follows:
Fig. 11: Overview of encrypting all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the positive bit array group.

Algorithm 18: Positive sentences of the training data set:
Input: ABiArray a bit array of one sentence
Output: a positive bit array group A Positive BitArray Group
Step 1: Receive ABiArray
Step 2: Add ABiArray into A Positive BitArray Group
Step 3: Return A Positive BitArray Group

From Fig. 12, we build the algorithm 19 and the algorithm 20 to encrypt all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the negative bit array group. This stage in Fig. 12 comprises two phases as follows: Hadoop Map phase and Hadoop Reduce phase. The input of the Hadoop Map is all the negative sentences of the training data set. The output of the Hadoop Map is ABiArray a bit array of one sentence. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is ABiArray a bit array of one sentence. The output of the Hadoop Reduce is a negative bit array group ANegativeBitArrayGroup.

We encrypt all the negative sentences of the training data set to negative the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the algorithm 19 to perform the Hadoop Map phase of this stage in the distributed environment, called the negative bit array group. The main ideas of the algorithm 19 are as follows:

Algorithm 19: The negative sentences of the training data set:
Input: all the negative sentences of the training data set
Output: ABiArray a bit array of one sentence
Step 1: Input all the negative sentences of the training data set into the Hadoop Map in the Cloudera system
Step 2: Each sentence in the negative sentences, do repeat
Step 3: ABiArray = the encoding one sentence in English to a binary array in the parallel system in Fig. 10 with the input is this sentence
Step 4: Return ABiArray

We encrypt all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the algorithm 20 to perform the Hadoop Reduce phase of this stage in the parallel system, called the negative bit array group. The main ideas of the algorithm 20 are as follows:
Algorithm 20: Negative sentences of the training data set to the bit arrays
Input: ABitArray a bit array of one sentence
Output: a negative bit array group A Negative Bit Array Group
Step 1: Receive ABitArray
Step 2: Add ABitArray into ANegativeBitArray Group
Step 3: Return ANegativeBitArray Group

From Fig. 13, we build the algorithm 21 and the algorithm 22 to transfer one document of the testing data set into the bit arrays of the document in the parallel system. This stage in Fig. 13 comprises two phases as follows: Hadoop Map phase and Hadoop Reduce phase. The input of the Hadoop Map is one document of the testing data set. The output of the Hadoop Map is one bit array of one sentence of the document—the output of the Hadoop Map. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is one bit array of one sentence of the document—the output of the Hadoop Map. The output of the Hadoop Reduce is the bit arrays of the document.

We propose the algorithm 21 to perform the Hadoop Map phase of transferring one document of the testing data set into the bit arrays of the document in the parallel system. The main ideas of the algorithm 21 are as follows:

Algorithm 21: Perform the Hadoop Map phase of transferring one document of the testing data set
Input: one document of the testing data set
Output: ABitArray one bit array of one sentence of the document the output of the Hadoop Map
Step 1: Input one document of the testing data set into the Hadoop Map
Step 2: Split this document into the sentences
Step 3: Each sentence in the sentences, do repeat
Step 4: ABitArray = the encoding one sentence in English to a binary array in the parallel system in Fig. 10 with the input is this sentence
Step 5: Return ABitArray

We propose the algorithm 22 to perform the Hadoop Reduce phase of transferring one document of the testing data set into the bit arrays of the document in the parallel system. The main ideas of the algorithm 22 are as follows:

Algorithm 22: Perform the Hadoop Reduce phase of transferring one document of the testing data set
Input: ABitArray one bit array of one sentence of the document the output of the Hadoop Map
Output: the bit arrays of the document The Bit Arrays of The Document
Step 1: Receive ABitArray
Step 2: Add ABitArray into The Bit Arrays of The Document
Step 3: Return TheBitArraysOfTheDocument
Fig. 13: Overview of transferring one document of the testing data set into the bit arrays of the document in the parallel system.

From Fig. 14, we build the algorithm 23 and the algorithm 24 to cluster one bit array (corresponding to one sentence) of the document into either the positive bit array group or the negative bit array group of the training data set by using the GA with the RWS in the distributed system. This stage in Fig. 15 comprises two phases as follows: Hadoop Map phase and Hadoop Reduce phase. The input of the Hadoop Map is one bit array (corresponding to one sentence) of the document, the positive bit array group and the negative bit array group the training data set. The output of the Hadoop Map is the bit array clustered into either the positive bit array group or the negative bit array group of the training data set. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is the bit array clustered into either the positive bit array group or the negative bit array group of the training data set. The output of the Hadoop Reduce is the bit array clustered into either the positive bit array group or the negative bit array group of the training data set.

We build the algorithm 23 to perform the Hadoop Map phase of clustering one bit array (corresponding to one sentence) of the document into either the positive bit array group or the negative bit array group of the training data set by using the GA with the RWS in the distributed system. The main ideas of the algorithm 23 are as follows:

**Algorithm 23: Perform the Hadoop Map phase of clustering one bit array (corresponding to one sentence):**

- **Input:** one bit array (corresponding to one sentence) of the document, the positive bit array group and the negative bit array group the training data set.
- **Output:** the bit array clustered into either the positive bit array group or the negative bit array group of the training data set.

**Step 1:** Input the bit array (corresponding to one sentence) of the document, the positive bit array group and the negative bit array group the training data set into the Hadoop Map in the Cloudera system.

**Step 2:** Randomly initialize population (1)

**Step 3:** Determine fitness of population (1)
Fig. 15: Overview of clustering one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the distributed environment

We build the algorithm 24 to perform the Hadoop Reduce phase of clustering one bit array (corresponding to one sentence) of the document into either the positive bit array group or the negative bit array group of the training data set by using the GA with the RWS in the parallel system. The main ideas of the algorithm 24 are as follows:

Algorithm 24: Perform the Hadoop Reduce phase of clustering one bit array:
Input: the bit array clustered into either the positive bit array group or the negative bit array group of the training data set

From Fig. 15, we build the algorithm 25 and the algorithm 26 to cluster one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the distributed environment. This stage in Fig. 15 comprises two phases as follows: Hadoop Map phase and Hadoop Reduce phase. The input of the Hadoop Map is one document of the testing data set; the positive bit array group and the negative bit array group of the training data set. The output of the Hadoop Map is OneResult the sentiment classification of one bit array of
the document. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is OneResult the sentiment classification of one bit array of the document. The output of the Hadoop Reduce is the sentiments (positive, negative or neutral) of the document.

We propose the algorithm 25 to perform the Hadoop Map phase of clustering one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the distributed environment. The main ideas of the algorithm 25 are as follows:

**Algorithm 25; The Hadoop Map phase of clustering one document of the testing data set:**

Input: one document of the testing data set; the positive bit array group and the negative bit array group of the training data set
Output: One Result the sentiment classification of one bit array of the document the output of the Hadoop Map
Step 1: Input the document of the testing data set; the positive bit array group and the negative bit array group of the training data set into the Hadoop Map in the Cloudera system
Step 2: AbitArray Group of One Document = the transferring one document of the testing data set into the bit arrays of the document in the parallel system in Fig. 13 with the input is this document
Step 3: Each bit array in AbitArray Group of One Document, do repeat
Step 4: One Result = the algorithm 10 to cluster one bit array (corresponding to one sentence) of the document into either the positive bit array group or the negative bit array group of the training data set by using the GA with the RWS in the sequential system with the input is this bit array, the positive bit array group and the negative bit array group of the training data set
Step 5: Return OneResult //the output of the Hadoop Map

We propose the algorithm 26 to perform the Hadoop Reduce phase of clustering one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the parallel environment. The main ideas of the algorithm 26 are as follows:

**Algorithm 26; Perform the Hadoop Reduce phase of clustering one document of the testing data set:**

Input: OneResult the sentiment classification of one bit array of the document the output of the Hadoop Map
Output: the sentiments (positive, negative or neutral) of the document
Step 1: Receive OneResult the sentiment classification of one bit array of the document
Step 2: If OneResult is the positive Then count_positive = count_positive + 1
Step 3: Else If OneResult is the negative Then count_negative = count_negative + 1
Step 4: If count_positive is greater than count_negative Then Return positive
Step 5: Else If count_positive is less than count_negative Then Return negative
Step 6: Return neutral

From Fig. 16, we build the algorithm 27 and the algorithm 28 to perform the Hadoop Map phase of clustering the documents of the testing data set into either the positive or the negative in the distributed environment. This stage in Fig. 16 comprises two phases as follows: Hadoop Map phase and Hadoop Reduce phase. The input of the Hadoop Map is 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 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 the testing data set. The output of the Hadoop Reduce is the results of the sentiment classification of the testing data set.

We build the algorithm 27 to perform the Hadoop Map phase of clustering the documents of the testing data set into either the positive or the negative in the distributed environment. The main ideas of the algorithm 27 area as follows:

**Algorithm 27; Perform the Hadoop Map phase of clustering the documents of the testing data set:**

Input: the testing data set and the training data set
Output: OneResult the result of the sentiment classification of one document the testing data set the output of the Hadoop Map
Step 1: The valences and the polarities of the sentiment lexicons of the bESD are calculated based on a basic English Sentiment Dictionary (bESD) in a distributed system
Step 2: A positive bit array group = encrypting all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the positive bit array group in Fig. 11 with the input is the positive sentences of the training data set
Step 3: A negative bit array group = encrypting all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the negative bit array group in Fig. 12 with the input is the positive sentences of the training data set
Step 4: Input the documents of the testing data set, the positive bit array group and the negative bit array group into the Hadoop Map in the Cloudera system
Step 5: Each document in the documents of the testing data set, do repeat
Step 6: OneResult = clustering one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the distributed environment in Fig. 15 with the input is this document, the positive bit array group and the negative bit array group
Step 7: Return OneResult

We build the algorithm 28 to perform the Hadoop Reduce phase of clustering the documents of the testing data set into either the positive or the negative in the parallel environment. The main ideas of the algorithm 28 area as follows:

**Algorithm 28; Perform the Hadoop Reduce phase of clustering the documents of the testing data set:**

Input: OneResult the result of the sentiment classification of one document the testing data set
Output: the results of the sentiment classification of the testing data set
Fig. 16: Overview of performing the Hadoop Map phase of clustering the documents of the testing data set into either the positive or the negative in the distributed environment

Step 1: Receive OneResult the result of the sentiment classification of one document the testing data set
Step 2: Add OneResult into The Results Of The Testing Dataset
Step 3: Return The Results of the Testing Dataset

RESULTS AND DISCUSSION

An Accuracy (A) is identified to calculate the accuracy of the results of the sentiment classification in this survey. We use a Java programming language programming to save data sets, implementing our proposed model to classify the 9,000,000 documents of the testing data set. To implement the proposed model, we have already used Java programming language to save the English testing data set and to save the results of the sentiment classification.

Our new model is performed in the sequential environment with the configuration as follows: The sequential environment in this research includes 1 node (1 server). The configuration of the server in the sequential environment is Intel® Server Board S1200V3RPS, Intel® Pentium® Processor G3220 (3M Cache, 3.00 GHz), 2GB CC3-10600 ECC 1333 MHz LP Unbuffered DIMMs. The operating system of the server is: Cloudera. The Java language is used in programming our model related to the Hamann
Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Function (FF).

We implement the proposed model related to the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Function (FF) in the Cloudera parallel network environment as follows: This Cloudera system includes 9 nodes (9 servers). The configuration of each server in the Cloudera system is: Intel® Server Board S1200V3RPS, Intel® Pentium® Processor G3220 (3M CHCHe, 3.00 GHz), 2GB CC3-10660 ECC 1333 MHz LP Unbuffered DDMs. The operating system of each server in the 9 servers is: Cloudera. All 9 nodes have the same configuration information. The Java language is used in programming the application of the proposed model related to the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Function (FF) in the Cloudera.

We show the significant information about this experiment of the proposed model in the table as follows: Table 5-7. Table 5, we show the results of the English documents in the testing data set. The accuracy of our new model for the English documents in the testing data set is presented in Table 6.

Table 7, we display the average times of the classification of our new model for the English documents in testing data set (Table 8-14).

Table 8: Average time of the classification of our new model for the English documents in testing data set

<table>
<thead>
<tr>
<th>Variables</th>
<th>Average time of the classification 9,000,000 English documents (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Function (FF) which is the Roulette-Wheel Selection (RWS) in the sequential environment</td>
<td>38,106.889</td>
</tr>
<tr>
<td>The Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Function (FF) which is the Roulette-Wheel Selection (RWS) in the Cloudera distributed system with 3 nodes</td>
<td>11,368.963</td>
</tr>
<tr>
<td>The Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Function (FF) which is the Roulette-Wheel Selection (RWS) in the Cloudera distributed system with 6 nodes</td>
<td>6,484.481</td>
</tr>
<tr>
<td>The Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Function (FF) which is the Roulette-Wheel Selection (RWS) in the Cloudera distributed system with 9 nodes</td>
<td>4,256.321</td>
</tr>
</tbody>
</table>

Table 9: Comparisons of our model's results with the works related to the Genetic Algorithm (GA) by Davis (1991), Kora and Krishna (2016), Yang et al. (2016), Etkaya and Uzunay (2016) and Wu et al. (2016)

<table>
<thead>
<tr>
<th>Studies</th>
<th>HC</th>
<th>CT</th>
<th>Sentiment classification</th>
<th>PNS</th>
<th>SD</th>
<th>DT</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Davis (1991)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>EL</td>
</tr>
<tr>
<td>Kora and Krishna (2016)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>EL</td>
</tr>
<tr>
<td>Yang et al. (2016)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>EL</td>
</tr>
<tr>
<td>Etkaya and Uzunay (2016)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>EL</td>
</tr>
<tr>
<td>Wu et al. (2016)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>EL</td>
</tr>
<tr>
<td>Our research</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>EL</td>
</tr>
</tbody>
</table>

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

Table 10: Comparisons of our model’s advantages and disadvantages with the researches related to the Genetic Algorithm (GA) by Davis (1991), Kora and Krishna (2016), Yang et al. (2016), Etkaya and Uzunay (2016) and Wu et al. (2016)

<table>
<thead>
<tr>
<th>Researches</th>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Davis (1991)</td>
<td>Genetic algorithms</td>
<td>This study is meant to be a practical guide for practitioners, not a textbook for a machine learning course. As a high-level introduction, the tutorial serves this purpose well and is strongly supplemented by the case studies. The survey is clearly written and enjoyable to read and short of insight. Davis himself is a consultant, reading his research is probably the quickest and easiest way to get off the ground for a first GA application</td>
<td>No mention</td>
</tr>
<tr>
<td>Kora and Krishna (2016)</td>
<td>Bundle block detection using differential evolution and Levenberg Marquardt neural network</td>
<td>The classification accuracy by the DE with LMNN was 99.1% for the detection of BSB. The proposed results have shown that the DE method can extract more relevant features than the other methods in the literature with highest classification accuracy for the detection of BSB</td>
<td>No mention</td>
</tr>
<tr>
<td>Yang et al. (2016)</td>
<td>A hybrid approach based on stochastic competitive Hopfield neural network and efficient Genetic algorithm for frequency assignment problem</td>
<td>In this survey, the researcher's first propose five optimal strategies to build an efficient Genetic algorithm. Then the researcher explore three hybridizations between SCHNN and EGA to discover the best hybrid algorithm. The researcher's believe that the comparison can also be helpful for hybridizations between neural networks and other evolutionary algorithms such as the particle swarm optimization algorithm, the artificial bee colony algorithm, etc. In the experiments, the researcher's hybrid algorithm obtains better or comparable performance than other algorithms on 5 benchmark problems and 12 large problems randomly generated. Finally, the researcher's show that the researcher's hybrid algorithm can obtain good results with a small size population</td>
<td>No mention</td>
</tr>
</tbody>
</table>
Table 10: Continue

<table>
<thead>
<tr>
<th>Researches</th>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Erkaya and Uzunay (2016)</td>
<td>Balancing of planar mechanisms having imperfect joints using neural Network-Based Genetic Algorithm (NN-GA) approach</td>
<td>In this study, dynamic response of mechanism having revolute joints with clearance is investigated. A four-bar mechanism having two revolute joints with clearance is considered as a model mechanism. Neural network was used to model several characteristics of joint clearance. Kinematic and dynamic analyses were achieved using continuous contact model between journal and bearing. A Genetic algorithm was also used to determine the appropriate values of design variables for reducing the additional vibration effects due primarily to the joint clearance. The results show that the optimum adjusting of suitable design variables gives a certain decrease in shaking forces and their moments on the mechanism frame.</td>
<td>No mention</td>
</tr>
<tr>
<td>Wu et al. (2016)</td>
<td>Genetic algorithm trajectory plan optimization for EAMA: EAST articulated maintenance arm, fusion engineering and design</td>
<td>This survey presents a trajectory optimization method which aims to pursue the 7-dof articulated arm a stable movement which keeps the mounted inspection camera anti-vibration. Based on dynamics analysis, trajectory optimization algorithm adopts multi-order polynomial interpolation in joint space and high order geometry Jacobian transform. The object of optimization algorithm is to suppress end-effector movement vibration by minimizing jerk, RMS (root mean square) value. The proposed solution has such characteristics which can satisfy kinematic constraints of EAMA’s motion and ensure the arm running under the absolute values of velocity, acceleration and jerk boundaries. GA (Genetic Algorithm) is employed to find global and robust solution for this problem.</td>
<td>No mention</td>
</tr>
</tbody>
</table>

Hamann Coefficient (HC) through the Google search engine with AND operator and OR operator, We use the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Function (FF) which is the Roulette-Wheel Selection (RWS) 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.

Table 11: Comparisons of our model’s results with the Roulette-Wheel Selection (RWS) by Panda et al. (2009), Lee et al. (1998), Tat and Tao (2003), Lipowski and Lipowska (2012) and Zou et al. (2006)

<table>
<thead>
<tr>
<th>Studies</th>
<th>HC</th>
<th>CT</th>
<th>Sentiment classification</th>
<th>PNS</th>
<th>SD</th>
<th>DT</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panda et al. (2009)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>EL</td>
</tr>
<tr>
<td>Lee et al. (1998)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>EL</td>
</tr>
<tr>
<td>Tat and Tao (2003)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>EL</td>
</tr>
<tr>
<td>Lipowski and Lipowska (2012)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>EL</td>
</tr>
<tr>
<td>Zou et al. (2006)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>EL</td>
</tr>
<tr>
<td>Our research</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>EL</td>
</tr>
</tbody>
</table>

Table 12: Comparisons of our model’s advantages and disadvantages with the Roulette-Wheel Selection (RWS) by Panda et al. (2009), Lee et al. (1998), Tat and Tao (2003), Lipowski and Lipowska (2012) and Zou et al. (2006)

<table>
<thead>
<tr>
<th>Researches</th>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panda et al. (2009)</td>
<td>PLANET: massively parallel learning of tree ensembles with MapReduce</td>
<td>In this survey, the researcher’s describe PLANET: a scalable distributed framework for learning tree models over large datasets. PLANET defines tree learning as a series of distributed computations and implements each one using the MapReduce Model of distributed computation. The researcher’s show how this framework supports scalable construction of classification and regression trees as well as ensembles of such models. The researcher’s discuss the benefits and challenges of using a MapReduce compute cluster for tree learning and demonstrate the scalability of this approach by applying it to a real world learning task from the domain of computational advertising.</td>
<td>No mention</td>
</tr>
<tr>
<td>Lee et al. (1998)</td>
<td>Digital filter design using Genetic algorithm</td>
<td>The researcher’s present an algorithm for designing 1-D FIR filters using genetic algorithms. In addition, the researcher’s examine the usefulness of various error norms such as 1/sum 2 / and 1/sum/sq/inf/(minimax) and their impact on the convergence rate and the optimal result</td>
<td>No mention</td>
</tr>
<tr>
<td>Tat and Tao (2003)</td>
<td>Using GIS and Genetic algorithm in highway alignment optimization</td>
<td>The Genetic Algorithm (GA) is a good choice as a search algorithm as its stochastic nature and global search characteristic enables the GA to find high quality solutions even for complex problems. The solution for an accurate cost model may lie with the use of the Geographic Information System (GIS). GIS can spatially represent both the physical, natural and socio-economic features of the region of the alignment. Furthermore, the spatial analytical capabilities of the GIS provide valuable inputs to the highway alignment optimization. This survey describes an integrated model that combines the capabilities of the GA and the GIS to optimize the highway alignments</td>
<td>No mention</td>
</tr>
<tr>
<td>Lipowski and Lipowska (2012)</td>
<td>Roulette-Wheel selection via stochastic acceptance</td>
<td>The researcher’s show that a Roulette-Wheel selection algorithm might be formulated as an algorithm of typically O(1) complexity. Previous implementations were of at least O(log n) complexity and were based on search methods. The researcher’s algorithm is based on a stochastic acceptance and is very simple what allows for its further modifications.</td>
<td>No mention</td>
</tr>
</tbody>
</table>
Hamann Coefficient (HC) through the Google search engine with AND operator and OR operator. We use the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Function (FF) which is the Roulette-Wheel Selection (RWS) 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.

Table 13: Comparisons of our model with the latest sentiment classification models (or the latest sentiment classification methods) by Agarwal and Mittal (2016a, b), Canuto et al. (2016), Ahmed and Danti (2016), Phu and Tuoi (2014), Ngoc et al. (2014), Dat et al. (2017), Phu et al. (2017a-d), Ngoc et al. (2017), Phu et al. (2017) and Phu et al. (2017a-h)

<table>
<thead>
<tr>
<th>Studies</th>
<th>Approach</th>
<th>Sentiment classification</th>
<th>PNS</th>
<th>SD</th>
<th>DT</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agarwal and Mittal</td>
<td>No</td>
<td>Yes</td>
<td>NM</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>(2016a)</td>
<td>No</td>
<td>Yes</td>
<td>NM</td>
<td>Yes</td>
<td>Yes</td>
<td>NM</td>
</tr>
<tr>
<td>Canuto et al. (2016)</td>
<td>No</td>
<td>Yes</td>
<td>NM</td>
<td>Yes</td>
<td>Yes</td>
<td>EL</td>
</tr>
<tr>
<td>Ahmed and Danti (2016)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Phu and Tuoi (2014)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>EL</td>
</tr>
<tr>
<td>Ngoc et al. (2014)</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>EL</td>
</tr>
<tr>
<td>Our research</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 14: Comparisons of our model’s positives and negatives the latest sentiment classification models (or the latest sentiment classification methods) by Agarwal and Mittal (2016a, b), Canuto et al. (2016), Ahmed and Danti (2016), Phu and Tuoi (2014), Ngoc et al. (2014), Dat et al. (2017), Phu et al. (2016), Ngoc et al. (2017), Phu et al. (2017) and Phu et al. (2017)

<table>
<thead>
<tr>
<th>Studies</th>
<th>Approach</th>
<th>Positives</th>
<th>Negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agarwal and Mittal</td>
<td>The machine learning approach 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 research in the following phases which are discussed in detail in this research for sentiment classification: feature extraction, feature weighting schemes, feature selection and machine-learning methods. This study also discusses the standard free benchmark datasets and evaluation methods for sentiment analysis. The researcher’s 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>
<tr>
<td>(2016a, b)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canuto et al. (2016)</td>
<td>Exploiting new sentiment-based meta-level features for effective Sentiment analysis</td>
<td>Experiments performed with a substantial number of datasets (nineteen) demonstrate that the effectiveness of the proposed sentiment-based meta-level features is notably superior to the traditional bag-of-words representation (up to 16%) but also is 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 researcher’s 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</td>
<td>A line of future research would be to explore the researcher’s 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</td>
</tr>
<tr>
<td>Ahmed and Danti (2016)</td>
<td>Rule-based machine learning algorithms</td>
<td>The proposed approach is tested by experimenting with online books and political reviews and demonstrates the efficiency 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 F1-score depicts higher efficiency rate and lower F1-score. Comparative experiments on various rule-based machine learning algorithms have been performed through a ten-fold cross validation training model for sentiment classification</td>
<td>No mention</td>
</tr>
<tr>
<td>Phu and Tuoi (2014)</td>
<td>The combination of term-counting method and enhanced contextual valence shifters method</td>
<td>The researcher’s 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</td>
<td>No mention</td>
</tr>
</tbody>
</table>
Table 14: Continue:

<table>
<thead>
<tr>
<th>Studies</th>
<th>Approach</th>
<th>Positives</th>
<th>Negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ngoc et al.</td>
<td>Naïve Bayes Model with N-gram method, negation handling method, Chi-square method and good-turing discounting, etc.</td>
<td>The researcher's have explored the Naïve Bayes Model with N-gram method, negation handling method, Chi-square method and good-turing discounting by selecting different thresholds of good-turing discounting method and different minimum frequencies of Chi-square method to improve the accuracy of sentiment classification</td>
<td>No mention</td>
</tr>
</tbody>
</table>

Hamann Coefficient (HC) through the Google search engine with AND operator and OR operator. We use the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Fuction (FF) which is the Roulette-Wheel Selection (RWS) 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.

CONCLUSION

A new model using the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) has been built to cluster the documents in English with Hadoop Map (M)/Reduce (R) in the Cloudera parallel network environment in this survey. With our proposed new model, we have achieved 88.02% accuracy of the testing data set as. 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.

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

We show the significant information about the average times of the sentiment classification of the proposed model.

The average time of the semantic classification of using the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) in the sequential environment is 38,106,889/9,000,000 sec English documents and it is greater than the average time of the emotion classification of using the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) in the Cloudera parallel network environment with 3 nodes which is 11,368,963/9,000,000 sec English documents.

The average time of the semantic classification of using the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Fuction (FF) in the sequential environment is the longest time in the table.

The average time of the emotion classification of using the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) in the Cloudera parallel network environment with 9 nodes which is 4,256,321/9,000,000 sec English documents is the shortest time in the table.

Besides, The average time of the emotion classification of using the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) in the Cloudera parallel network environment with 6 nodes is 6,484,481/9,000,000 sec English documents.

The average time of the emotion classification of using the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) in the Cloudera parallel network environment with 3 nodes is greater than the average time of the emotion classification of using the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) in the Cloudera parallel network environment with 6 nodes.

The execution time of using the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) 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 execution time of the proposed model depends on many factors as follows:
The GA related algorithms
The Hamann coefficient related algorithms
The performance of the distributed network environment
The performance of each node of the distributed environment
The performance of each server of the parallel system
The number of the nodes of the parallel environment
The testing data set and the training data set
The sizes of the data sets
The parallel functions such as Hadoop Map and Hadoop Reduce
The operating system of the parallel network such as the Cloudera

The proposed model has many advantages and disadvantages. Its positives are as follows: It uses the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) and the Fitness Function (FF) to classify the 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. It can save a lot of storage space. 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 discussed in this study.

We show the comparisons of our model’s results with the studies related to the Genetic Algorithm (GA) by Davis (1991), Kora and Krishna (2016), Yang et al. (2016), Erkaya and Uzmay (2016), Wu et al. (2016). The comparisons of our model’s advantages and disadvantages with the studies related to the genetic algorithm (GA) by Davis (1991), Kora and Krishna (2016), Yang et al. (2016), Erkaya and Uzmay (2016), Wu et al. (2016).

We display the comparisons of our model’s results with the Roulette-Wheel Selection (RWS) by Panda et al. (2009), Lee et al. (1998), Tat and Tao (2003), Lipowski and Lipowska (2012), Zou et al. (2006).

The comparisons of our model’s advantages and disadvantages with the Roulette-Wheel Selection (RWS) by Panda et al. (2009), Lee et al. (1998), Tat and Tao (2003), Lipowski and Lipowska (2012), Zou et al. (2006).

We present the comparisons of our model with the latest sentiment classification models (or the latest sentiment classification methods) by Agarwal and Mittal (2016 ab), Canuto et al. (2016), Ahmed and Danti (2016), Phu and Tuoi (2014), Ngoc et al. (2017), Dat et al. (2017), Phu et al. (2016, 2017 a-h).

The comparisons of our model’s positives and negatives the latest sentiment classification models (or the latest sentiment classification methods) by Agarwal and Mittal (2016 ab), Canuto et al. (2016), Ahmed and Danti (2016), Phu and Tuoi (2014), Ngoc et al. (2017), Dat et al. (2017), Phu et al. (2017 a-h).

Appendices
Appendix of code
Algorithm 1: Creating a basis English Sentiment Dictionary (bESD) in a sequential environment.

Algorithm 2: Implementing the Hadoop Map phase of creating a basis English Sentiment Dictionary (bESD) in a distributed environment.

Algorithm 3: Performing the Hadoop Reduce phase of creating a basis English Sentiment Dictionary (bESD) in a distributed environment.

Algorithm 4: Encrypting one sentiment lexicon (comprising the content and the valence) to a binary array in the sequential environment.

Algorithm 5: Encoding one sentence in English to a binary array in the sequential system.

Algorithm 6: Encrypting all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bsd in the sequential system.

Algorithm 7: Encoding all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bsd in the sequential system.

Algorithm 8: Transferring one document of the testing data set into the bit arrays of the document in the sequential system.

Algorithm 9: Genetic algorithm in the sequential environment.

Algorithm 10: Clustering one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the sequential environment.
Algorithm 11: Clustering the documents of the testing data set into either the positive or the negative in the sequential environment.

Algorithm 12: Implementing the Hadoop Map phase of encrypting one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment.

Algorithm 13: Performing the Hadoop Reduce phase of encrypting one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment.

Algorithm 14: Performing the Hadoop Map phase of encoding one sentence in English to a binary array in the parallel system.

Algorithm 15: Implementing the Hadoop Reduce of encoding one sentence in English to a binary array in the parallel system.

Algorithm 16: Encrypting all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD to perform the Hadoop Map phase of this stage in the distributed environment, called the positive bit array group.

Algorithm 17: Encrypting all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD to perform the Hadoop Reduce phase in the parallel system, called the positive bit array group.

Algorithm 18: Encrypting all the negative sentences of the training data set to negative the bit arrays based on the bit arrays of the sentiment lexicons of the bESD to perform the Hadoop Map phase in the distributed environment, called the negative bit array group.

Algorithm 19: Encrypting all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD to perform the Hadoop Reduce phase in the parallel system, called the negative bit array group.

Algorithm 20: Performing the Hadoop Map phase of transferring one document of the testing data set into the bit arrays of the document in the parallel system.

Algorithm 21: Performing the Hadoop Reduce phase of transferring one document of the testing data set into the bit arrays of the document in the parallel system.

Algorithm 22: Performing the Hadoop Map phase of clustering one bit array (corresponding to one sentence) of the document into either the positive bit array group or the negative bit array group of the training data set by using the GA with the RWS in the distributed system.

Algorithm 23: Performing the Hadoop Reduce phase of clustering one bit array (corresponding to one sentence) of the document into either the positive bit array group or the negative bit array group of the training data set by using the GA with the RWS in the parallel system.

Algorithm 24: Performing the Hadoop Map phase of clustering one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the distributed environment.

Algorithm 25: Performing the Hadoop Reduce phase of clustering one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the parallel environment.

Algorithm 26: Performing the Hadoop Map phase of clustering the documents of the testing data set into either the positive or the negative in the distributed environment.

Algorithm 27: Performing the Hadoop Reduce phase of clustering the documents of the testing data set into either the positive or the negative in the parallel environment.

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

Algorithm 2: Implementing the Hadoop Map phase of creating a basis English Sentiment Dictionary (bESD) in a distributed environment:
Input: the 55,000 English terms; the Google search engine
Output: one term which the sentiment score and the polarity are identified
Begin
Step 1: Each term in the 55,000 terms, do repeat
Step 2: By using Eq. 8-10 of the calculating a valence of one word (or one phrase) in English in the study, the sentiment score and the polarity of this term are identified. The valence and the polarity are calculated by using the HC through the Google search engine with AND operator and OR operator
Step 3: Return this term
End

Algorithm 3: Performing the Hadoop Reduce phase of creating a basic English Sentiment Dictionary (bESD) in a distributed environment:
Input: one term which the sentiment score and the polarity are identified. The output of the Hadoop Map phase
Output: a basic English Sentiment Dictionary (bESD)
Begin
Step 1: Receive this term
Step 2: Add this term into the basic English Sentiment Dictionary (bESD)
Step 3: Return bESD
End

Algorithm 4: Encrypting one sentiment lexicon (comprising the content and the valence) to a binary array in the sequential environment:
Input: one sentiment lexicon of the bESD
Output: a bit array
Begin
Step 1: Split this term into the letters
Step 2: Set AbitArray = null
Step 3: Set Valence = Get a valence of this term based on the bESD
Step 4: Each letter in the letters, do repeat
Step 5: Based on the binary code of letters in English by Anonymous (2017 g-h), we get a bit array of this letter
Step 6: Add the bit array of this letter into AbitArray
Step 7: End Repeat-End Step 3
Step 8: Based on the transferring a decimal to a binary code by Anonymous (2017 m-r), we transfer the valence to a bit array
Step 9: Add this bit array into AbitArray
Step 10: Return AbitArray
End

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

Algorithm 6: Encrypting all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bad in the sequential system:
Input: all the positive sentences of the training data set
Output: a positive bit array group
Begin
Step 1: Set APositiveBitArrayGroup = null
Step 2: Each sentence in the positive sentences, do repeat
Step 3: AbitArray = the algorithm 5 to encode one sentence in English to a binary array in the sequential system with the input is this sentence
Step 4: Add AbitArray into APositiveBitArrayGroup
Step 5: End Repeat-End Step 2
Step 6: Return APositiveBitArrayGroup
End

Algorithm 7: Encoding all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bad in the sequential system
Input: all the negative sentences of the training data set
Output: a negative bit array group
Begin
Step 1: Set ANegativeBitArrayGroup = null
Step 2: Each sentence in the positive sentences, do repeat
Step 3: AbitArray = the algorithm 5 to encode one sentence in English to a binary array in the sequential system with the input is this sentence
Step 4: Add AbitArray into ANegativeBitArrayGroup
Step 5: End Repeat-End Step 2
Step 6: Return ANegativeBitArrayGroup
End

Algorithm 8: Transferring one document of the testing data set into the bit arrays of the document in the sequential system:
Input: one document of the testing data set
Output: the bit arrays of the document
Begin
Step 1: Set TheBitArraysOfTheDocument = null
Step 2: Split this document into the sentences
Step 3: Each sentence in the sentences, do repeat
Step 4: AbitArray = the algorithm 5 to encode one sentence in English to a binary array in the sequential system with the input is this sentence
Step 5: Add AbitArray into The Bit Arrays of the Document
Step 6: End Repeat-End Step 3
Step 7: Return TheBitArraysOfTheDocument
End

Algorithm 9: Genetic algorithm in the sequential environment:
Input:
P: the input data set includes the binary bit sequences of the Algorithm 3 (the binary data set table) //initial//population
Output
P’ /improved population and it is the information to build decision tree
Begin
1. Repeat
2. N = |P|
3. P’ = {} 4. Repeat
5. i1, i2 = select (P, Fitness)
6. o1, o2 = cross (i1, i2, Fitness)
7. o1 = mutate (o1, Fitness)
8. o2 = mutate (o2, Fitness)
9. P’ = P’ + /o1, o2/
10. until |P’| = N
11. P = P’
12. Until termination criteria satisfied, // the best individual (or the) in P,
   according to Fitness (the individual has the //highest Fitness
End

Algorithm 10: clustering one document of the testing data set into either the positive bit array group or the
negative bit array group of the training data set using the GA with the RWS in the sequential environment:
Input: one document of the testing data set; the positive bit array group and the negative bit array group of the training data set
Output: the sentiments (positive, negative or neutral)
Begin
Step 1: AbitArray GroupOfOneDocument: = the algorithm 8 to transfer one document of the testing data set into the bit arrays of the document in the sequential system with the input is this document
Step 2: count_positive := 0 and count_negative := 0
Step 3: Each bit array in AbitArray Group Of One Document, do repeat:
Step 4: One Result: = the algorithm 9 to cluster one bit array (corresponding to one sentence) of the document into either the positive bit array group or the negative bit array group of the training data set by using the GA with the RWS in the sequential system with the input is this bit array, the positive bit array group and the negative bit array group of the training data set
Step 5: If One Result is the positive Then count_positive := count_positive + 1
Step 6: Else If One Result is the negative then count_negative := count_negative + 1
Step 7: End Repeat – End Step 3
Step 8: If count_positive is greater than count_negative Then return positive
Step 9: Else If count_positive is less than count_negative Then return negative
Step 10: Return neutral
End

Algorithm 11; Clustering the documents of the testing data set into either the positive or the negative in the sequential environment:
Input: the testing data set and the training data set
Output: the results of the sentiment classification of the testing data set
Begin
Step 1: The valences and the polarities of the sentiment lexicons of the bESD are calculated based on a basis English Sentiment Dictionary (bESD) in a sequential environment
Step 2: A positive bit array group: = encrypt all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the algorithm 6 with the input is the positive sentences of the training data set
Step 3: A negative bit array group: = encode all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the algorithm 7 with the input is the positive sentences of the training data set
Step 4: Set The Results Of The Testing Data Set := null
Step 5: Each document in the documents of the testing data set, do repeat
Step 6: One Result := the algorithm 10 to cluster one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the sequential environment with the input is this document, the positive bit array group and the negative bit array group
Step 7: Add One Result into The Results Of The Testing Data Set
Step 8: End Repeat – End Step 5
Step 9: Return The Results Of The Testing Data Set
End

Algorithm 12; Implementing the hadoop map phase of encrypting one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment:
Input: one sentiment lexicon of the bESD
Output: a bit array of one letter
Begin
Step 1: Input this term and the bESD into the Hadoop Map in the Cloudera system
Step 2: Split this term into the letters
Step 3: Set Valence := Get a valence of this term based on the bESD
Step 4: Each letter in the letters, do repeat
Step 5: Based on the binary code of letters in English by Anonymous (2017g), we get a bit array of this letter
Step 6: Return the bit array of this letter; //the output of the Hadoop Map
End

Algorithm 13; Performing the Hadoop Reduce phase of encrypting one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment:
Input the bit array of this letter; //the output of the Hadoop Map
Output a bit array of the term-ABitArray
Begin
Step 1: Receive the bit array of this letter
Step 2: Add the bit array of this letter into ABitArray
Step 3: If this term is full then
Step 4: Based on the transferring a decimal to binary code by Anonymous (2017h), we transfer the valence to a bit array
Step 5: Add this bit array into ABitArray
Step 6: End If – End Step 3
Step 7: Return ABitArray
End

Algorithm 14; Performing the Hadoop Map phase of encoding one sentence in English to a binary array in the parallel system:
Input: one sentence
Output: a bit array of one term ABitArray
Begin
Step 1: Input this sentence into the Hadoop Map in the Cloudera system
Step 2: Split this sentence into the meaningful terms (meaningful words or meaningful phrase)
Step 3: Each term in the terms, do repeat
Step 4: AbitArray: = the encrypting one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment in Fig. 12 with the input is this term
Step 5: Return ABitArray
End

Algorithm 15; Implementing the Hadoop Reduce of encoding one sentence in English to a binary array in the parallel system:
Input: a bit array of one term “ABitArray” the output of the Hadoop Map
Output: a bit array of the sentence ABitArray of Sentence
Begin
Step 1: Receive ABitArray
Step 2: Add ABitArray into ABitArray of Sentence
Step 3: Return ABitArray of Sentence
End

Algorithm 16; Encrypting all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD to perform the Hadoop Map phase of this stage in the distributed environment, called the positive bit array group:
Input: all the positive sentences of the training data set
Output: ABitArray—a bit array of one sentence
Begin
Step 1: Input all the positive sentences of the training data set into the Hadoop Map in the Cloudera system
Step 2: Each sentence in the positive sentences, do repeat:
Step 3: ABitArray: = the encoding one sentence in English to a binary array in the parallel system in Fig. 13 with the input is this sentence
Step 4: Return ABitArray
End
Algorithm 17: Encrypting all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD to perform the Hadoop Reduce phase in the parallel system, called the positive bit array group:
Input: ABitArray—a bit array of one sentence
Output: positive bit array group APositiveBitArray Group
Begin
Step 1: Receive ABitArray
Step 2: Add ABitArray into APositiveBitArrayGroup
Step 3: Return APositiveBitArrayGroup
End

Algorithm 18: Encrypting all the negative sentences of the training data set to the negative bit arrays based on the bit arrays of the sentiment lexicons of the bESD to perform the Hadoop Map phase in the distributed environment, called the negative bit array group:
Input: all the negative sentences of the training data set
Output: ABitArray a bit array of one sentence
Begin
Step 1: Input all the negative sentences of the training data set into the Hadoop Map in the Cloudera system
Step 2: Each sentence in the negative sentences, do repeat
Step 3: ABitArray = the encoding one sentence in English to a binary array in the parallel system in Fig. 13 with the input is this sentence
Step 4: Return ABitArray
End

Algorithm 19: Encrypting all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD to perform the Hadoop Reduce phase in the parallel system, called the negative bit array group:
Input: ABitArray—a bit array of one sentence
Output: negative bit array group ANegativeBitArray Group
Begin
Step 1: Receive ABitArray
Step 2: Add ABitArray into ANegativeBitArrayGroup
Step 3: Return ANegativeBitArrayGroup
End

Algorithm 20: Performing the Hadoop Map phase of transferring one document of the testing data set into the bit arrays of the document in the parallel system:
Input: one document of the testing data set
Output: ABitArray one bit array of one sentence of the document the output of the Hadoop Map
Begin
Step 1: Input one document of the testing data set into the Hadoop Map in the Cloudera system
Step 2: Split this document into the sentences
Step 3: Each sentence in the sentences, do repeat
Step 4: ABitArray = the encoding one sentence in English to a binary array in the parallel system in Fig. 13 with the input is this sentence
Step 5: Return ABitArray
End

Algorithm 21: Performing the Hadoop Reduce phase of transferring one document of the testing data set into the bit arrays of the document in the parallel system:
Input: ABitArray one bit array of one sentence of the document the output of the Hadoop Map
Output: the bit arrays of the document TheBitArrays Of The Document
Begin
Step 1: Receive ABitArray
Step 2: Add ABitArray into TheBitArrays Of The Document
Step 3: Return TheBitArrays Of The Document
End

Algorithm 22: Performing the Hadoop Map phase of clustering one bit array (corresponding to one sentence) of the document into either the positive bit array group or the negative bit array group of the training data set by using the GA with the RWS in the distributed system:
Input: one bit array (corresponding to one sentence) of the document, the positive bit array group and the negative bit array group the training data set
Output: the bit array clustered into either the positive bit array group or the negative bit array group of the training data set
Begin
Step 1: Input the bit array (corresponding to one sentence) of the document the positive bit array group and the negative bit array group the training data set into the Hadoop Map in the Cloudera system
Step 2: randomly initialize population (t)
Step 3: determine fitness of population (t)
Step 4: repeat
Step 5: select parents from population (t)
Step 6: perform crossover on parents creating population(t+1)
Step 7: perform mutation of population(t+1)
Step 8: determine fitness of population (t+1)
Step 9: until best individual is good enough
Step 10: Return this bit array clustered into either the positive bit array group or the negative bit array group of the training data set
End

Algorithm 23: Performing the Hadoop Reduce phase of clustering one bit array (corresponding to one sentence) of the document into either the positive bit array group or the negative bit array group of the training data set by using the GA with the RWS in the parallel system:
Input: the bit array clustered into either the positive bit array group or the negative bit array group of the training data set
Output: the sentiments (positive, negative or neutral)
Begin
Step 1: Receive the bit array clustered into either the positive bit array group or the negative bit array group of the training data set
Step 2: If this bit array clustered into the positive bit array group Then Return positive
Step 3: If this bit array clustered into the negative bit array group Then Return negative
Step 4: Return neutral
End

Algorithm 24: Performing the Hadoop Map phase of clustering one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the distributed environment:
Input: one document of the testing data set; the positive bit array group and the negative bit array group of the training data set
Output: One result the sentiment classification of one bit array of the document the output of the Hadoop Map
Begin
Step 1: Input the document of the testing data set; the positive bit array group and the negative bit array group of the training data set into the
Algorithm 25: Performing the Hadoop Reduce phase of clustering one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the parallel environment:
Input: OneResult the result of the sentiment classification of one document
Output: the positive and negative (neutral) of the document
Begin
Step 1: Receive OneResult the result of the sentiment classification of one bit array of the document
Step 2: If OneResult is the positive Then count_positive := count_positive + 1
Step 3: Else If OneResult is the negative Then count_negative := count_negative + 1
Step 4: If count_positive is greater than count_negative Then Return positive
Step 5: Else If count_positive is less than count_negative Then Return negative
Step 6: Return neutral
End

Algorithm 26: Performing the Hadoop Map phase of clustering the documents of the testing data set into either the positive or the negative in the distributed environment:
Input: the testing data set and the training data set
Output: OneResult the result of the sentiment classification of one document
Begin
Step 1: The valences and the polarities of the sentiment lexicons of the bESD are calculated based on a basic English Sentiment Dictionary (bESD) in a distributed system
Step 2: A positive bit array group = the encrypting all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the positive bit array group in Fig. 14 with the input is the positive sentences of the training data set
Step 3: A negative bit array group = the encrypting all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the negative bit array group in Fig. 15 with the input is the negative sentences of the training data set
Step 5: Each document in the documents of the testing data set, do repeat:
Step 6: OneResult = the clustering one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the distributed environment in Fig. 17 with the input is this document, the positive bit array group and the negative bit array group
Step 7: Return OneResult
End

Algorithm 27: Performing the Hadoop Reduce phase of clustering the documents of the testing data set into either the positive or the negative in the parallel environment:
Input: OneResult the result of the sentiment classification of one document the testing data set
Output: the result of the sentiment classification of the testing data set
Begin
Step 1: Receive OneResult the result of the sentiment classification of one document the testing data set
Step 2: Add OneResult into the result of the TestingDataSet
Step 3: Return The result of the TestingDataSet
End

RECOMMENDATIONS

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


