Journal of Engineering and Applied Sciences 14 (7): 2055-2094, 2019
ISSN: 1816-949X

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

[1]Vo Ngoc Phu and [2]Vo Thi Ngoc Tran
[1]Nguyen Tat Thanh University, 300A Nguyen Tat Thanh Street, Ward 13, District 4, 702000 Ho Chi Minh City, Vietnam
[2]School 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

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 Fuction (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 (bESD) 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 bESD 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 bESD 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

- Millions of English documents are successfully processed for emotional analysis
- The semantic classification is implemented in the parallel network environment
- The principles are proposed in the research
- The Cloudera distributed environment is used in this study
- The proposed study can be applied to other distributed systems
- This survey uses Hadoop Map (M) and Hadoop Reduce (R)
- Our proposed model can be applied to many different parallel network environments such as a Cloudera system
- This study can be applied to many different distributed functions such as Hadoop Map (M) and Hadoop Reduce (R)
- The GA-related algorithms are built in this survey
- The HC-related algorithms are proposed in this study

Literature review: We summarize many researches which are related to our research. By far, we know that PMI (Pointwise Mutual Information) equation and SO (Sentiment Orientation) equation are used for determining polarity of one word (or one phrase) and strength of sentiment orientation of this word (or this phrase). Jaccard Measure (JM) is also used for calculating polarity of one word and the equations from this Jaccard measure are also used for calculating strength of sentiment orientation this word in other research. PMI, Jaccard, Cosine, Ochiai, Tanimoto and Sorensen measure are the similarity measure between two words from those, we prove that the Hamann Coefficient (HC) is also used for identifying valence and polarity of one English word (or one English phrase). Finally, we identify the sentimental values of English verb phrases based on the basis English semantic lexicons of the basis English Emotional Dictionary (bESD).

There are the studies related to PMI measure by Bai *et al.* (2014), Turney and Littman (2002), Malouf and Mullen (2017), Scheible (2010), Jovanoski *et al.* (2015), Htait *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) and Wang and Araki (2007). In the research of Bai *et al.* (2014), the researchers generate several Norwegian sentiment lexicons by extracting sentiment information from two different types of Norwegian text corpus, namely, news corpus and discussion forums. The methodology is based on the Point wise Mutual Information (PMI). The researchers introduce a modification of the PMI that considers small "blocks" of the text instead of the text as a whole. The study by Turney and Littman (2002) introduces a simple algorithm for unsupervised learning of semantic orientation from extremely large corpora, etc.

Two studies related to the PMI measure and Jaccard measure are 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), Omar *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 concernabout 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), Schnyer *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), Erkaya and Uzmay, 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 ECG using optimized features from the ECG beats. For the detection of normal and BBB beats, these DE feature values are given as the input for the LMNN classifier, etc.

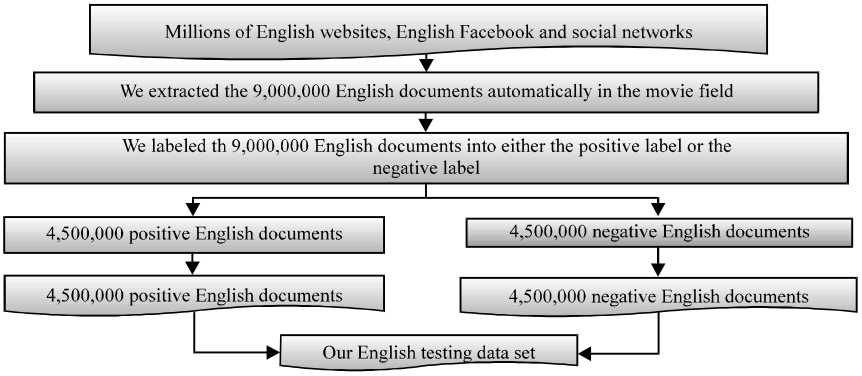


Fig. 1: Our English training data set

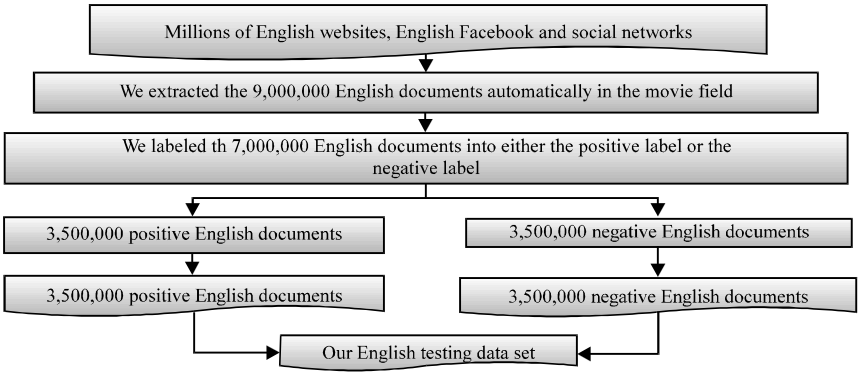


Fig. 2: Our English training data set

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 fo 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 ab), Canuto *et al.* (2016), Ahmed and Danti (2016), Phu and Tuoi (2014), Ngoc *et al.* (2017), Dat *et al.* (2017), Phu *et al.* (2017a-h). In the research of Agarwal and Mittal (2016a, 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 (2016a, b) discusses an approach where an exposed stream of tweets from the Twitter micro blogging site are pre-processed and classified based on their sentiments. In sentiment classification system the concept of opinion subjectivity has been accounted. In the study, the 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 the first part, we identify a sentiment value of one word (or one phrase) in English. In the the 2nd part, we create a basis English Sentiment Dictionary (bESD) in a sequential system. In the 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), Htait *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:

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

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

$$\mathrm{SO}\left(w_i\right) = \mathrm{PMI}\left(w_i, \text{positive}\right) - \mathrm{PMI}\left(w_i, \text{negative}\right) \quad (2)$$

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

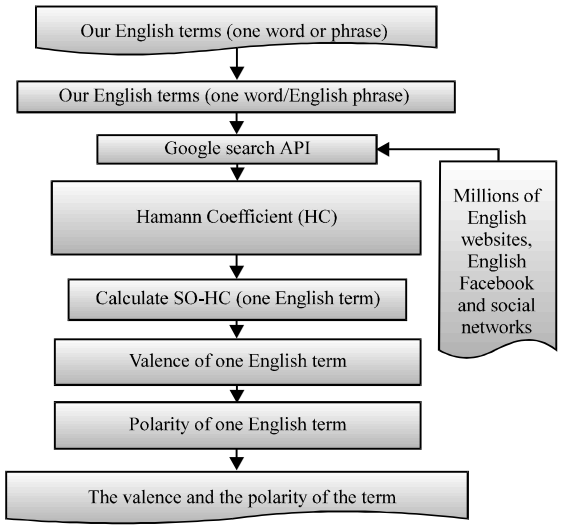


Fig. 3: Overview of identifying the valence and the polarity of one term in English using a Hamann Coefficient (HC)

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), Htait *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 Htait *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 Htait *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 Chinesese 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 researches 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:

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

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

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

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

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

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), Jovanoski *et al.* (2015), Htait *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), 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), Ponomarenko *et al.* (2002), Meyer *et al.* (2004), Mladenovic *et al.* (2008), Tamas *et al.* (2001) and Phu *et al.* (2017 a-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:

$$\text{Hamann coefficient}(a, b) = \text{Hamann measure}(a, b) = HC(a, b) = \frac{\left[(a \cap b)+(\neg a \cap \neg b)-(\neg a \cap \neg b)+(a \cap \neg b)\right]}{(a \cap b)+(\neg a \cap b)+(a \cap \neg b)+(\neg a \cap \neg b)} \quad (6)$$

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_1 and b is replaced by w_2 Eq. 7.

$$\text{Hamann measure}(w_1, w_2) =$$
$$\text{Hamann coefficient}(w_1, w_2) =$$
$$HC(w_1, w_2) = \frac{\left[P(w_1, w_2)+P(\neg w_1, \neg w_2)\right]-\left[P(\neg w_1, \neg w_2)+P(\neg w_1, \neg w_2)\right]}{\left[P(w_1, w_2)+P(\neg w_1, \neg w_2)\right]+\left[P(w_1, \neg w_2)+P(\neg w_1, \neg w_2)\right]} \quad (7)$$

Equation 7 is similar to 1. Equation 2, 1 is replaced by Eq. 7. We have Eq. 8 as follows:

$$\text{Valence}(w) = SO_{HC(w)} = HC(w, \text{positive}_{query}) - HC(w, \text{negative}_{query}) \quad (8)$$

Equation 7 shows w_1 is replaced by w and w_2 is replaced by position_query. We have Eq. 9 is as follows:

$$A9 = \left[P(w, \text{positive}_{query})+P(\neg w, \neg \text{positive}_{query})\right]$$
$$B9 = \left[P(\neg w, \text{positive}_{query})+P(w, \neg \text{positive}_{query})\right] \quad (9)$$
$$C9 = P(w, \text{positive}_{query})+P(\neg w, \text{positive}_{query})+P(w, \neg \text{positive}_{query})+P(\neg w, \neg \text{positive}_{query})$$

Equation 7 shows w_1 is replaced by w and w_2 is replaced by negative_query. We have Eq. 10. Equation 10 is as follows:

$$HC(w, \text{negative}_{query}) = \frac{A10 - B10}{C10} \quad (10)$$

With:

$$A10 = \left[P(w, \text{negative}_{query})+P(\neg w, \neg \text{negative}_{query})\right]$$
$$B10 = \left[P(\neg w, \text{negative}_{query})+P(w, \neg \text{negative}_{query})\right] \quad (11)$$
$$C9 = P(w, \text{negative}_{query})+P(\neg w, \text{negative}_{query})+P(w, \neg \text{negative}_{query})+P(\neg w, \neg \text{negative}_{query})$$

Algorithm 1; w_1, w_2, w_3 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_1, w_2): number of returned results in Google Search by keyword (w_1 and w_2). We use the Google Search API to get the number of returned results in search online Google by keyword (w_1 and w_2)

3) P(w_1): number of returned results in Google Search by keyword w_1. We use the Google Search API to get the number of returned results in search online Google by keyword w_1

4) P(w_2): number of returned results in Google Search by keyword w_2. We use the Google Search API to get the number of returned results in search online Google by keyword w_2

5) Valence(W) = SO-HC(w): valence of English word (or English phrase) w is SO of word (or phrase) by using the Hamann Coefficient (HC)

6) Positive_query: {HCtive or good or positive or beautiful or strong or nice or excellent or fortunate or correct or superior} with the positive query is the a group of the positive English words

7) Negative_query: {passive or bad or negative or ugly or week or nasty or poor or unfortunate or wrong or inferior} with the negative_query is the a group of the negative English words

8) P (w, positive_query): number of returned results in Google Search by keyword (positive_query and w). We use the Google Search API to get the number of returned results in search online Google by keyword (positive_query and w)

9) P(w, negative_query): number of returned results in Google Search by keyword (negative_query and w). We use the Google Search API to get the number of returned results in search online Google by keyword (negative_query and w)

10) P(w): number of returned results in Google Search by keyword w. We use the Google Search API to get the number of returned results in search online Google by keyword w

11) P(¬w,positive_query): number of returned results in Google Search by keyword ((not w) and positive_query). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and positive_query)

12) P(w, ¬positive_query): number of returned results in the Google Search by keyword (w and (not (positive_query))). We use the Google Search API to get the number of returned results in search online Google by keyword (w and [not (positive_query)])

13) P(¬w, ¬positive_query): number of returned results in the Google Search by keyword (w and (not (positive_query))). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and [not (positive_query)])

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

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

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

According to Cosine, Ochiai, Sorensen, Tanimoto, PMI and Jaccard about calculating the valence (score) of the word, we identify the valence (score) of the English word w based on both the proximity of positive_query with w and the remote of positive_query with w and the proximity of negative_query with w and the remote of negative_query with w.

If HC (w, positive_query) is as equal as 1, the English word w is the nearest of positive_query

If HC(w, positive_query) is as equal as 0, the English word w is the farthest of positive_query

If HC (w, positive_query) >0 and HC (w, positive_query) = 1, the English word w belongs to positive_query being the positive group of the English words

If HC (w, negative_query) is as equal as 1, the English word w is the nearest of negative_query

If HC (w, negative_query) is as equal as 0, the English word w is the farthest of negative_query

If HC (w, negative_query)>0 and HC (w, negative_query) = 1, the English word w belongs to negative_query being the negative group of the English words.

So, the valence of the English word w is the value of HC (w, positive_query) substracting the value of HC (w, negative_query) and the Eq. 8 is the equation of identifying the valence of the English word w. We have the information about HC as follows:

1) HC (w, positive_query) = 0 and HC (w, positive_query) = 1

2) HC (w, negative_query) = 0 and HC (w, negative_query) = 1

3) If HC (w, positive_query) = 0 and HC (w, negative_query) = 0 then SO_HC (w) = 0

4) If HC (w, positive_query) = 1 andHC (w, negative_query) = 0 then SO_HC (w) = 0

5) If HC (w, positive_query) = 0 and HC (w, negative_query) = 1 then SO_HC(w) = -1

6) If HC (w, positive_query) = 1 and HC (w, negative_query) = 1 then SO_HC(w) = 0

So, SO_HC (w) = -1 and SO_HC (w)≤1

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 negative 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).

In Table 1, we shows the comparisons of our model's results with the studies related to Bai *et al.* (2014), Turney and Littman (2002), Malouf and Mullen (2017), Scheible (2010), Jovanoski *et al.* (2015), Htait *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), 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), Ponomarenko *et al.* (2002), Meyer *et al.* (2004), Mladenovic *et al.* (2008), Tamas *et al.* (2001) and Phu *et al.* (2017 a-h).

The comparisons of our model's advantages and disadvantages with the studies related to Bai *et al.* (2014), Turney and Littman (2002), Malouf and Mullen (2017), Scheible (2010), Jovanoski *et al.* (2015), Htait *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), 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), Ponomarenko *et al.* (2002), Meyer *et al.* (2004), Mladenovic *et al.* (2008), Tamas *et al.* (2001) and Phu *et al.* (2017 a-h) are displayed in Table 2.

Table 1: The sentiment lexicons of the bESD

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

In Table 3, we present the comparisons of our model's results with the studies srelated to the Hamann Coefficient (HC) by Choi *et al.* (2010), Schraw (1995), David *et al.* (1959), Schnyer *et al.* (2004), Grigsby *et al.* (1998) and Urbina *et al.* (2010).

The 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), Schnyer *et al.* (2004), Grigsby *et al.* (1998) and Urbina *et al.* (2010) are displayed in Table 4.

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: the 55,000 English terms; the Google Search engine
Output: a basis English Sentiment Dictionary (bESD)
Step 1: Each term in the 55,000 terms, do repeat
Step 2: By using Eq. 8-10 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 2: Comparisons of our model's results with the researches related to Bai *et al.* (2014), Turney and Littman (2002), Malouf and Mullen (2017), Scheible (2010), Jovanoski *et al.* (2015), Htait *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), 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), Ponomarenko *et al.* (2002), Meyer *et al.* (2004), Mladenovic *et al.* (2008), Tamas *et al.* (2001) and (Phu *et al.* (2017a-h)

Studies	PMI	JM	Language	SD	DT	HC	SC	Other measures	Search engines
Bai *et al.* (2014)	Yes	No	English	Yes	Yes	No	Yes	No	No Mention
Turney and Littman (2002)	Yes	No	English	Yes	No	No	Yes	Latent Semantic Analysis (LSA)	AltaVista
Malouf and Mullen (2017)	Yes	No	English	Yes	Yes	No	Yes	Baseline; Turney-inspired; NB; Cluster+NB; Human	AltaVista
Scheible (2010)	Yes	No	English German	Yes	Yes	No	Yes	SimRank	Google search engine
Jovanoski *et al.* (2015)	Yes	No	English Macedonian	Yes	Yes	No	Yes	No Mention	AltaVista search engine
(Htait *et al.* (2016)	Yes	No	English Arabic	Yes	No	No	Yes	No Mention	Google search engine Bing search engine
Wan (2009)	Yes	No	English Chinese	Yes	Yes	No	Yes	SVM(CN); SVM(EN); SVM(ENCN1); SVM(ENCN2); TSVM(CN); TSVM(EN); TSVM(ENCN1); TSVM(ENCN2); CoTrain	No mention
Brooke *et al.* (2009)	Yes	No	English Spanish	Yes	Yes	No	Yes	SO Calculation SVM	Google
Jiang *et al.* (2015)	Yes	No	Chinese Tibetan	Yes	Yes	No	Yes	Feature selection Expectation Cross Entropy Information Gain	No mention
Tan and Zhang (2008)	Yes	No	Chinese	Yes	Yes	No	Yes	DF, CHI, MI andIG	No mention
Du *et al.* (2010)	Yes	No	Chinese	Yes	No	No	Yes	Information Bottleneck Method (IB); LE	AltaVista
Zhang *et al.* (2010)	Yes	No	Chinese	Yes	Yes	No	Yes	SVM	Google, Yahoo, Baidu
Wang and Araki (2007)	Yes	No	Japanese	No	No	No	Yes	Harmonic-Mean	Google and replaced the Near operator with the and operator int he SO formula
Feng *et al.* (2013)	Yes	Yes	English	Yes	Yes	No	Yes	Dice; NGD	Google search engine
Nguyen and Hagiwara	Yes	Yes	English	Yes	No	No	Yes	Dice; Overlap	Google
Shikalgar and Dixit (2014)	No	Yes	English	Yes	Yes	No	Yes	A Jaccard Index Based Clustering Algorithm (JIBCA)	No mention
Ji *et al.* (2015)	No	Yes	English	Yes	Yes	No	Yes	Naive Bayes, Two-Step Multinomial Naive Bayes and Two-Step Polynomial-Kernel Support Vector Machine	Google
Omar *et al.* (2013)	No	Yes	Arabic	No	No	No	Yes	Naive Bayes (NB); Support Vector Machines (SVM); Rocchio; Cosine	No mention
Omar *et al.* (2013)	No	Yes	Chinese	Yes	Yes	No	Yes	A new score-Economic Value (EV), etc.	Chinese search
Ren *et al.* (2014)	No	Yes	Chinese	Yes	Yes	No	Yes	Cosine	No mention
Netzer *et al.* (2012)	No	Yes	English	No	Yes	No	Yes	Cosine	No mention
Ren *et al.* (2011)	No	Yes	Chinese	No	Yes	No	Yes	Dice; overlap; Cosine	No mention
Phu *et al.* (2017a-h)	No	No	Vietnamese	No	No	No	Yes	Ochiai measure	Google
Phu *et al.* (2017a-h)	No	No	English	No	No	No	Yes	Cosine coefficient	Google
Phu *et al.* (2017a-h)	No	No	English	No	No	No	Yes	Sorensen measure	Google
Phu *et al.* (2017a-h)	No	Yes	Vietnamese	No	No	No	Yes	Jaccard	Google
Phu *et al.* (2017a-h)	No	No	English	No	No	No	Yes	Tanimoto coefficient	Google
Our research	No	No	English Language	No	No	Yes	Yes	No	Google Search engine

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 isone 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 built 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 3: Comparisons of our model's advantages and disadvantages with the researches related to Bai *et al.* (2014), Turney and Littman (2002), Malouf and Mullen (2017), Scheible (2010), Jovanoski *et al.* (2015), Htait *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), 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), Ponomarenko *et al.* (2002), Meyer *et al.* (2004), Mladenovic *et al.* (2008), Tamas *et al.* (2001), Phu *et al.* (2017a-h)

Surveys	Approach	Advantages	Disadvantages
Bai *et al.* (2014)	Constructing sentiment lexicons in Norwegian from a large text corpus	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 lengths that generate better sentiment lexicons. Some of the researcher's preliminary research showed that 100 gave a better result	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
Turney and Littman (2002)	Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus	This survey has presented a general strategy for learning semantic orientation from semantic association, SO-A. Two instances of this strategy have been empirically evaluated SO-PMI-IR andSO-LSA. The accuracy of SO-PMI-IR is comparable to the accuracy of HM, the algorithm of Hatzivassiloglou and McKeown. SO-PMI-IR requires a large corpus but it is simple,easy to implement, unsupervised and it is not restricted to adjectives.	No mention
Malouf and Mullen (2017)	Graph-based user classification for informal online political discourse	The researcher's describe several experimentsin 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	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
Scheible (2010)	Anovel, graph-based approach using SimRank	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	The researcher's future research will include a further examination ofthe merits of its application for knowledge-sparse languages
Jovanoski *et al.* (2015)	Analysis in Twitter for Macedonian	The researcher's experimental results show an F1-score of 92.16 which is very strong and is on par with the best results for English which were achieved in recent SemEval competitions	In future research, the researcher's are interested in studying the impact of the raw corpus size, e.g., the researcher's could onlycollect half a million Tweets for creating lexicons and analyzing/evaluating the system while Kiritchenko *et al.* built their lexicon on million Tweets and evaluated their system on 135 million English Tweets. Moreover, the researcher's are interested notonly in quantity but also in quality, i.e. in studying the quality of the individual words and phrasesused as seeds
Htait *et al.* (2016)	Using web search engines for English and Arabic unsupervised sentiment intensity prediction	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	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
Wan (2009)	Co-training for cross-lingual sentiment classification	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	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 adopted for employ sentiment classification. The researcher's will the Structural Correspondence Learning domain adaption algorithm used by Blitzer (SCL) *et al.* for linking the translated text and the natural text
Brooke *et al.* (2009)	Cross-linguistic sentiment analysis: From English to Spanish	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	No mention

Table 3: Continue

Surveys	Approach	Advantages	Disadvantages
Jiang *et al.* (2015)	Micro-blog emotion orientation analysis algorithm based on Tibetan and Chinese mixed text	By emotion orientation analyzing and studying of Tibetan microblog which is concerned in Sina, making Tibetan Chinese emotion dictionary, Chinese sentences, Tibetan part of speech sequence and emotion symbol as emotion factors and using expected cross entropy combined fuzzy set to do feature selection to realize a kind of microblog emotion orientation analyzing algorithm based on Tibetan and Chinese mixed text. The experimental results showed that the method can obtain better performance in Tibetan and Chinese mixed Microblog orientation analysis	No mention
Tan and Zhang (2008)	An empirical study of sentiment analysis for Chinese documents	Four feature selection methods (MI, IG, CHI and DF) and five learning methods (centroid classifier, k-nearest neighbor, winnow classifier, Naive Bayes and SVM) are investigated on a Chinese sentiment corpus with a size of 1021 documents. The experimental results indicate that IG performs the best for sentimental terms selection and SVM exhibits the best performance for sentiment classification. Furthermore, the researcher's found that sentiment classifiers are severely dependent on domains or topics	No mention
Du *et al.* (2010)	Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon	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 proposed method greatly outperforms the baseline methods in the task of building out-of-domain sentiment lexicon	In this study, only the mutual information measure is employed to measure the three kinds of relationship. In order to show the robustness of the framework, the researcher's future effort is to investigate how to integrate more measures into this framework
Zhang *et al.* (2010)	Sentiment classification for consumer word-of-mouth in Chinese: comparison between supervised and unsupervised approaches	This study adopts three supervised learning approaches and a web-based semantic orientation approach, PMI-IR, to Chinese reviews. The results show that SVM outperforms naive bayes and N-gram model on various sizes of training examples but does not obviously exceeds the semantic orientation approach when the number of training examples is smaller than 300	No mention
Wang and Araki (2007)	Modifying SO-PMI for Japanese weblog opinion mining by using a balancing factor and detecting neutral expressions	After these modifications, the researcher's chieved a well-balanced result: both positive and negative accuracy exceeded 70%. 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.	In the future, the researcher's will evaluate different choices of wordsfor the sets of positive and negative reference words. The researcher's also plan to appraise their proposal on other languages
Feng *et al.* (2013)	In this survey, the researcher's empirically evaluatethe performance of different corpora in sentiment similarity measurement which is the fundamental task for word polarity classification	Experiment results show that the Twitter data can Achieve a much better performance than the Google, Web1T and Wikipedia based methods	No mention
Nguyen and Hagiwara	Adjective-based estimation of short sentence's impression	The adjectives are ranked and top na adjectives are considered as an output of system. For example, the experiments were carried out and got fairly good results. With the input "it is snowy", the results are white (0.70), light (0.49), cold (0.43), solid (0.38) and scenic (0.37)	In the 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
Shikalgar and Dixit (2014)	Jaccard index based clustering algorithm for mining online review	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	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 websites, etc.
Ji *et al.* (2015)	Twitter sentiment classification for measuring public health concerns	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)	no mention

Table 3: Continue

Surveys	Approach	Advantages	Disadvantages
Omar *et al.* (2013)	Ensemble of classification algorithms for subjectivity and sentiment analysis of arabic customer's reviews	The experimental results show that the ensemble of the classifiers improves the classification effectiveness in terms of macro-F1 for both levels The best results obtained from the subjectivity analysis and the sentiment classification in terms of macro-F1 are 97.13 and 90.95%, respectively	No mention
Mao *et al.* (2014)	Automatic construction of financial semantic orientation lexicon from large-scale Chinese News Corpus	Semantic orientation lexicon of positive and negative words is indispensable for sentiment analysis. However, many lexicons are manually created by a small number of human subjects which are susceptible to high cost and bias. In this survey, the researcher's propose a novel idea to construct a financial semantic orientation lexicon from large-scale Chinese news corpus automatically	No mention
Ren *et al.* (2014)	Sentiment classification in under- resourced languages using graph-based semi-supervised learning methods	In particular, the researcher's found that choosing initially labeled vertices in HC cordance with their degree and PageRank score can improve the performance. However, pruningunreliable 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	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 planto exploit a much larger amount of unlabeled data to fully take advantage of SSL algorithms
Netzer *et al.* (2012)	A text-mining approach and combine it with semantic network analysis tools	In summary, the researcher's hope the text-mining and derivedmarket-structure analysis presented in this study provides a first step in exploring the extremely large, rich and useful body of consumer data readily availableon Web 2.0	No mention
Ren *et al.* (2011)	Sentiment classification in resource-scarce languagesby using label propagation	The researcher's compared our method with supervised learning and semi-supervised learning methods onreal Chinese reviews classification in three domains. Experimental results demonstrated that label propagation showed a competitive performance against SVM or Transductive SVM with besthyper-parameter settings. Considering the difficulty of tuning hyper-parameters in a resource scarce setting, the stable performance of parameter-free label propagation is promising	The researcher's plan to further improve the performance of LP in sentiment classification, especially when the researcher's only have a small number of labeled seeds. The researcher's will exploit the idea of restricting the label propagating steps when the available labeled data is quite small.
Phu *et al.* (2017a-h)	A Vietnamese adjective emotion dictionary based on exploitation of Vietnamese language characteristics	The Vietnamese adjectives often bear emotion which values (or semantic scores) are not fixed and are changed when they appear indifferent contexts of these phrases. Therefore, if the Vietnamese adjectives bring sentimentand their semantic values (or their sentiment scores) are not changed in any context, thenthe results of the emotion classification are not high accuracy. The researcher's propose many rulesbased on Vietnamese language characteristics to determine the emotional values of theVietnamese 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.	Not calculating all Vietnamese words completely; not identifying all Vietnamese adjective phrases fully, etc.
Phu *et al.* (2017a-h)	A valences-totaling model for English sentiment classification	The researcher's present a full range of English sentences, thus, theemotion 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	It has low accuracy, it misses manysentiment-bearing English words; it misses many sentiment-bearing English phrases because sometimes the valence of a English phrase is not the total of the valences of the English wordsin this phrase; it misses many English sentences which are not processed fully and it misses many English documents which are not processed fully
Phu *et al.* (2017a-h)	Shifting semantic values of English phrases for classification	The results of the sentiment classification are not high accuracy if the English phrases bring the emotions and their semantic values (or their sentiment scores) are not changed in any context. For those reasons, the 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	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.

Table 3: Continue

Surveys	Approach	Advantages	Disadvantages
		The results of this research are widely used in applications and researches of the English semantic classification	
Phu *et al.* (2017a-h)	A valence-totaling model for Vietnamese sentiment classification	The researcher's have used the VTMfV to classify 30,000 Vietnamese documents which include the 15,000 positive Vietnamese documents and the 15,000 negative Vietnamese documents. The researcher's have achieved accuracy in 63.9% of the researcher's' Vietnamese testing data set VTMfV is not dependent on the special domain VTMfV is also not dependent on the training data set and there is no training stage in this VTMfV. From the researcher's results in this research, our VTMfV can be applied in the different fields of the Vietnamese natural language processing. In addition, the researcher's TCMfV can be applied to many other languages such as Spanish, Korean, etc. It can also be applied to the big data set sentiment classification in Vietnamese and can classify millions of the Vietnamese documents	It has a low accuracy
Phu *et al.* (2017a-h)	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 Englis 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: 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 advantages and disadvantages of this survey are shown in the Conclusion section

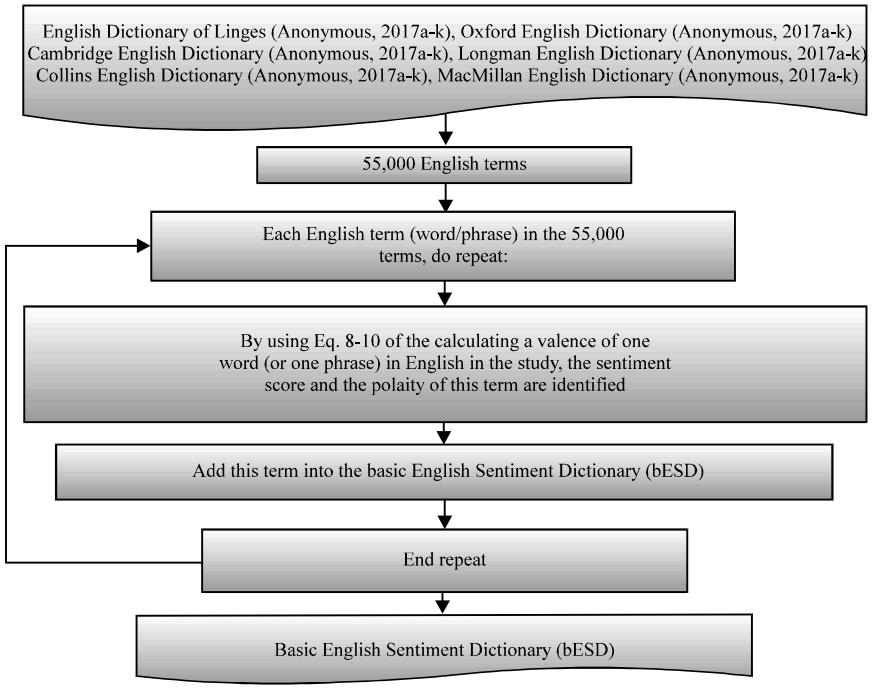


Fig. 4: Overview of creating a basis English Sentiment Dictionary (bESD) in a sequential environment

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), Schnyer *et al.* (2004), Grigsby *et al.* (1998) and Urbina *et al.* (2010)

Studies	PMI	JM	Hamann Coefficient (HC)	Language	SD	DT	Sentiment classification
Choi *et al.* (2010)	Yes	Yes	Yes	English	NM	NM	No mention
Schraw (1995)	No	No	Yes	NM	NM	NM	No mention
David *et al.* (1959)	No	No	Yes	NM	NM	NM	No mention
Schnyer *et al.* (2004)	No	No	Yes	NM	NM	NM	No mention
Grigsby *et al.* (1998)	No	No	Yes	NM	NM	NM	No mention
Urbina *et al.* (2010)	No	No	Yes	NM	NM	NM	No mention
Our research	No	No	Yes	English Language	Yes	Yes	Yes

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), Schnyer *et al.* (2004), Grigsby *et al.* (1998) and Urbina *et al.* (2010)

Surveys	Approach	Benefits	Drawbacks
Choi *et al.* (2010)	A survey of binary similarity and distance measures	Applying appropriate measures results in more accurate data analysis. Notwithstanding, few comprehensive surveys on binary measures have been conducted. Hence the researcher's collected 76 binary similarityand distance measures used over the last century and reveal their correlations through the hierarchical clustering technique	No mention
Schraw (1995)	Measures of feeling-of-knowing accuracy: A new look at an old problem	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 as 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	No mention
David *et al.* (1959)	The second virial coefficients of some cyclic hydrocarbons	This survey reports some measurements of the second virial coefficient B of cyclopropane in the temperature range 300-400°K. 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 650°K	No mention
Schnyer *et al.* (2004)	A role for right medial prefrontal cortex in accurate feeling-of-knowing judgments: evidence from patients with lesions to frontal cortex	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.	No mention
Grigsby *et al.* (1998)	Executive cognitive abilities and functional status among community-dwelling older persons in the san luis valley health and aging study	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	No-mention
Urbina *et al.* (2010)	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	Subjects with T1DM had higher body mass index, LDL-cholesterol, fasting glucose and blood pressure than control subjects. Diabetic subjects had lower BrachD and higher AIx-75 indicating increased stiffness. Age-adjusted pulse wave velocity-trunk (aorto-femoral) was higher in cases (all $p \leq 0.05$) Increased peripheral stiffness was more common than central stiffness in subjects with T1DM (low BrachD in 33% vs high PWV-trunk 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	No mention

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 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 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

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

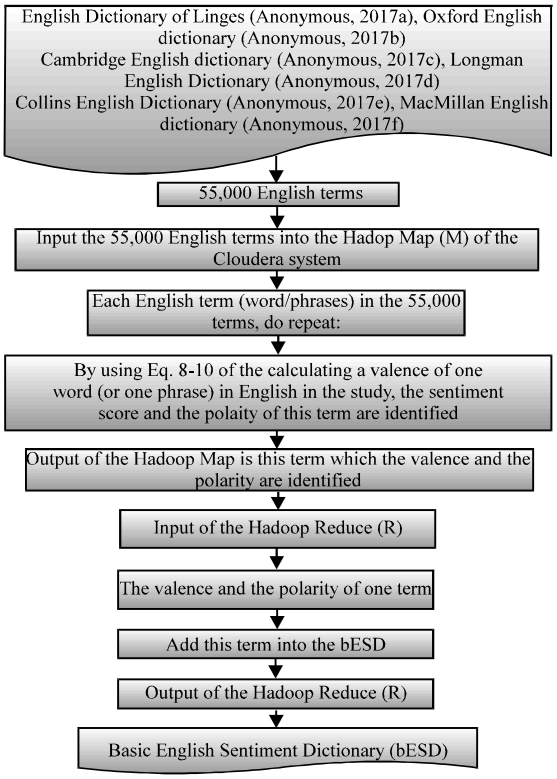


Fig. 5: Overview of creating a basis English Sentiment Dictionary (bESD) in a distributed environment

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 study compises 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.

Table 6: The results of the English documents in the testing data set

Variables	Testing dataset	Correct classification	Incorrectclassification
Negative	4,500.000	3,970.864	00529.136
Positive	4,500.000	3,950.936	00549.064
Summary	9,000.000	7,921.800	1,078.200

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 lexicons 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 (2017 a-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 (2017 a-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 evolved 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

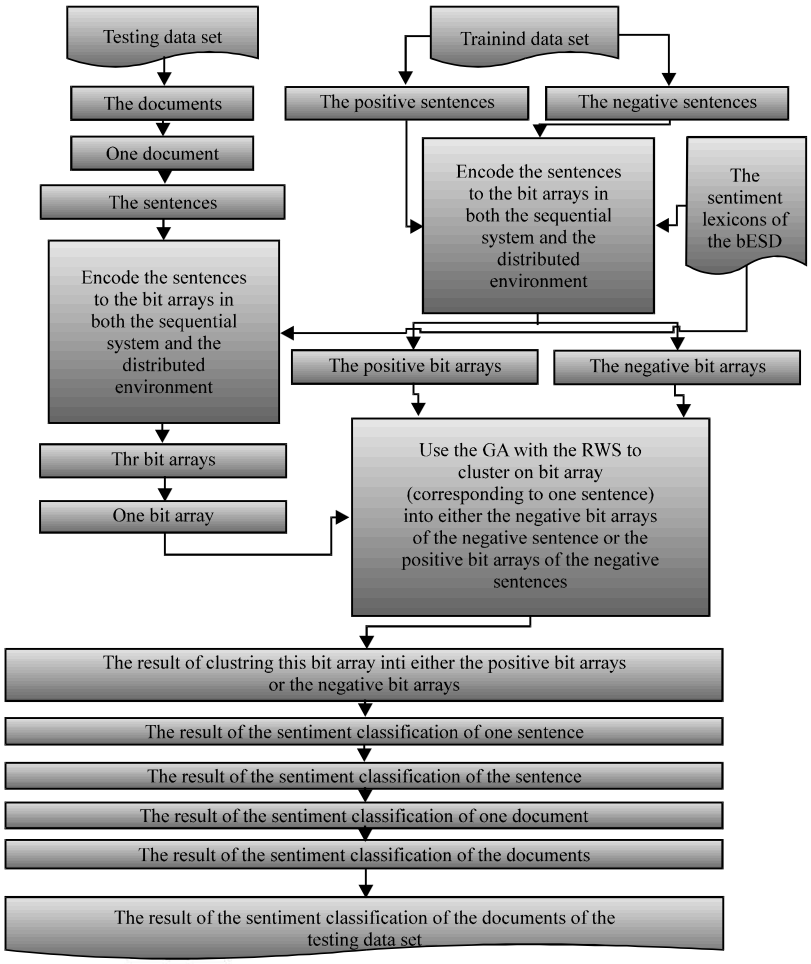


Fig. 6: Overview of implementing the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) in both a sequential environment and a distributed network system

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 fti which is defined by:

$$f_{ti} = \sum_{j=1}^{i} f_i$$

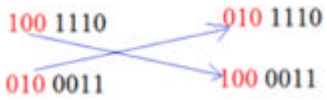
Call Fn is the sum of the adaptive coefficient in all the population. Pick a random number f between 0 and Fn. Select the first instance correspond f = ftk into new population.

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 newchromosomes into the population. Example:



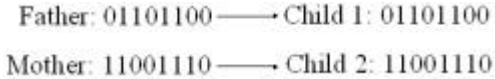
Mutation operator (mutation): Mutation is a phenomenon which the child chromosomes carry some features not in the genetic code of the father-mother chromosomes.

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

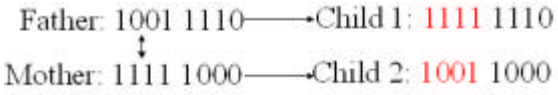
Example:

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

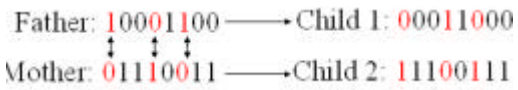
Asexual reproduction: Each child is an exact copy of each father or each mother. Example:



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:



Example of the sexual reproduction intersecting 3 bits:



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 the 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), Tat 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 spinned. 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 theRoulette-Wheel Selection (RWS) the fitness fuction (FF) in a sequential environment: 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 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:

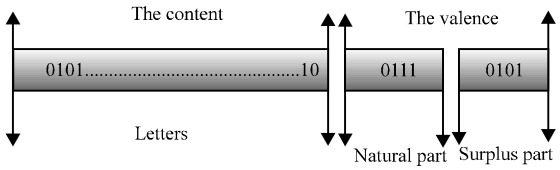


Fig. 7: Overview of presenting one sentiment lexicon of the bESD in a binary code

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-k), we get a bit array of this letter
Step 6: Add the bit array of this letter into ABitArray
Step 7: End Repeat
End Step 3
Step 8: Based on on the transferring a decimal to a binary code 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

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 ABitArrayOfSentence

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 APositiveBitArrayGroup
Step 5: End Repeat-End Step 2
Step 6: Return APositiveBitArrayGroup

We encode all the negative sentences of the training data set to the negative bit arrays based on the bit arrays

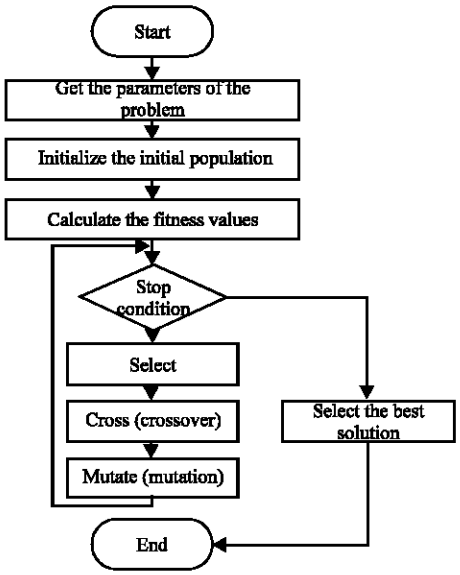


Fig. 8: The diagram of the GA in the sequential environment

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 A NegativeBitArray Group
Step 5: End Repeat-End Step 2
Step 6: Return A Negative BitArray 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 Th eBit Arrays 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 Bit Arrays 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 groupand 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: = 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 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:

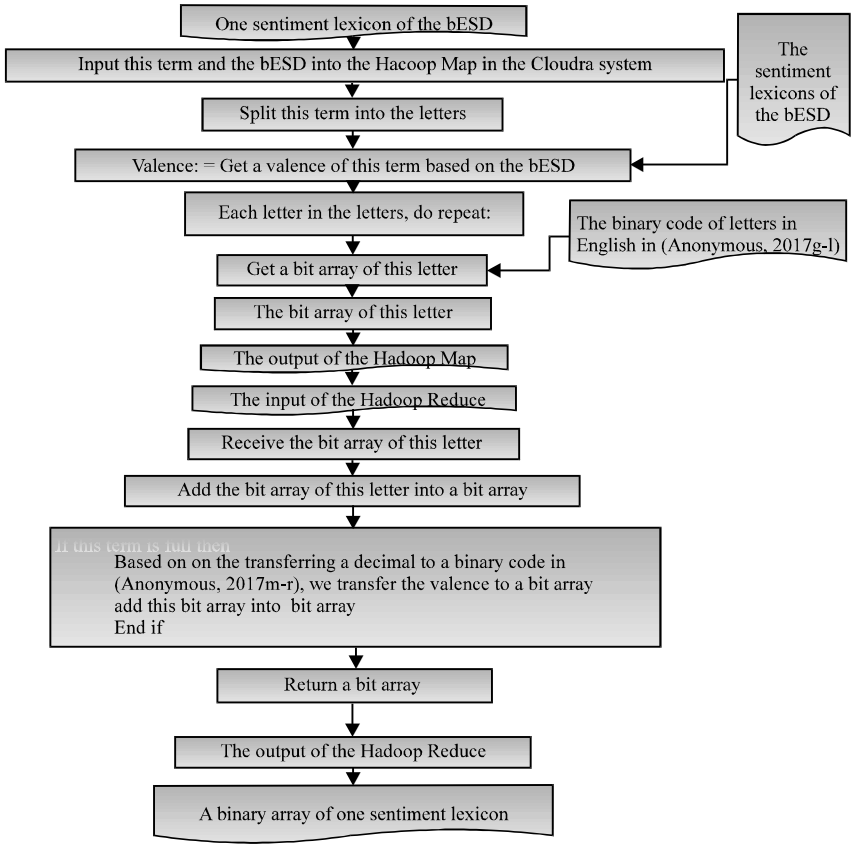


Fig. 9: Overview of encrypting one sentiment lexicon (comprising the content and the valence) to a binary array in the distributed network environment

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 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 (2017 g-l), 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 by Anonymous (2017 m-r), 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 isa 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

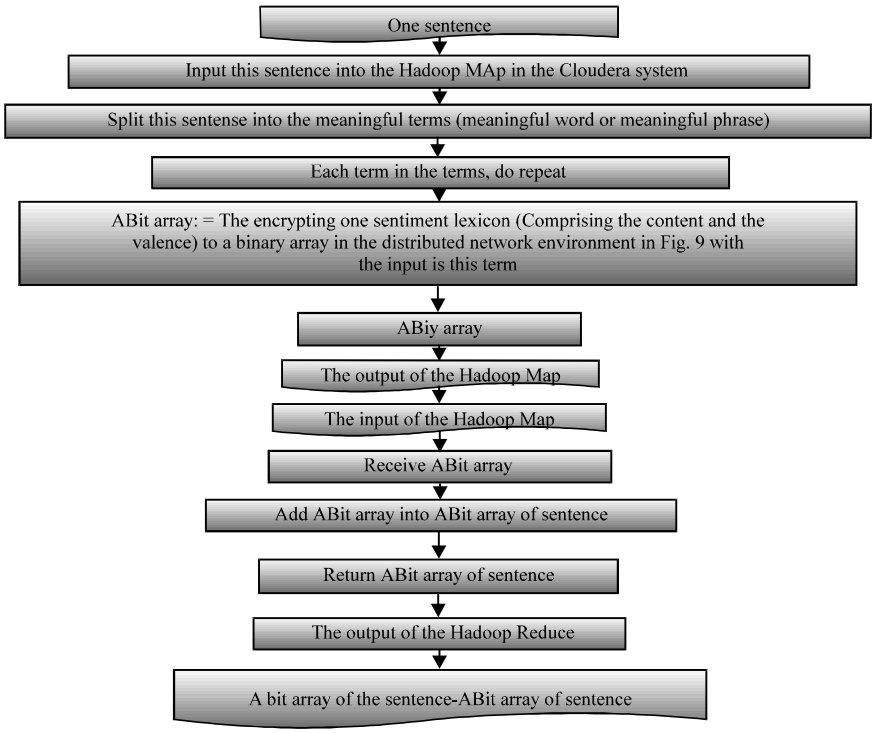


Fig. 10: Overview of encoding one sentence in English to a binary array in the parallel system

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. 11 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 Englis:

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:

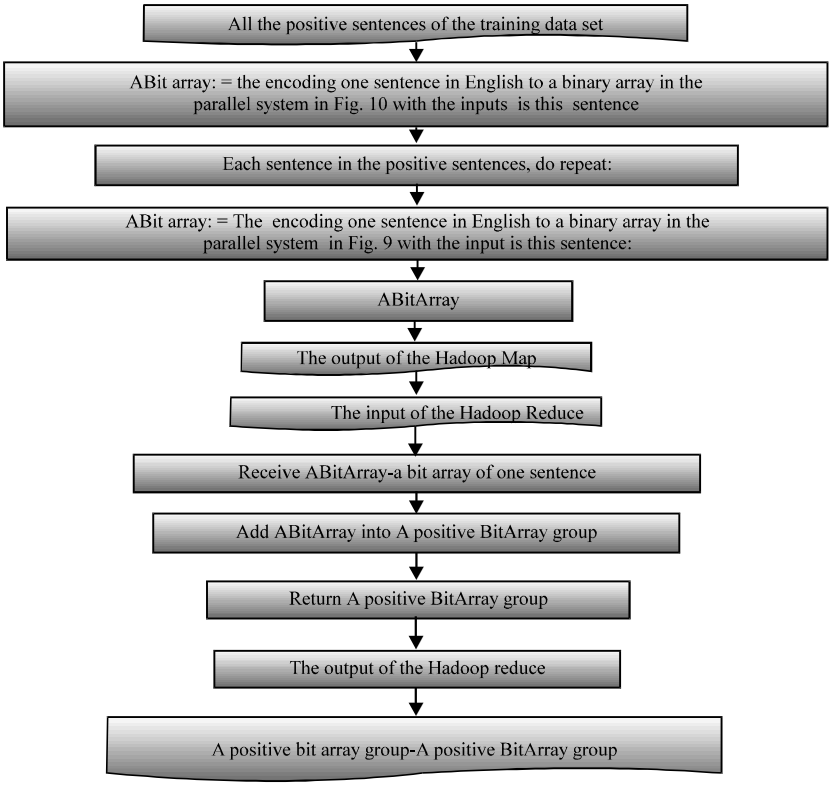


Fig. 11: Overview of encrypting all the positive sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the positive bit array group

Algorithm 18; Positive sentences of the training data set:

Input:ABitArray a bit array of one sentence
Output:a positive bit array group A Positive BitArray Group
Step 1: Receive ABitArray
Step 2: Add ABitArray into Apositive BitArray Group
Step 3: Return APositiveBitArrayGroup

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 ABitArray a bit array of one sentence. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is ABitArray a bit array of one sentence. The output of the Hadoop Reduce is a negative bit array group 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:ABitArray a bit array of one sentence
Step 1: Input all the negative sentences of the training data set into the Hadoop Map in the Cloudera system
Step 2: Each sentence in the negative sentences, do repeat
Step 3: AbitArray: = the 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 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:

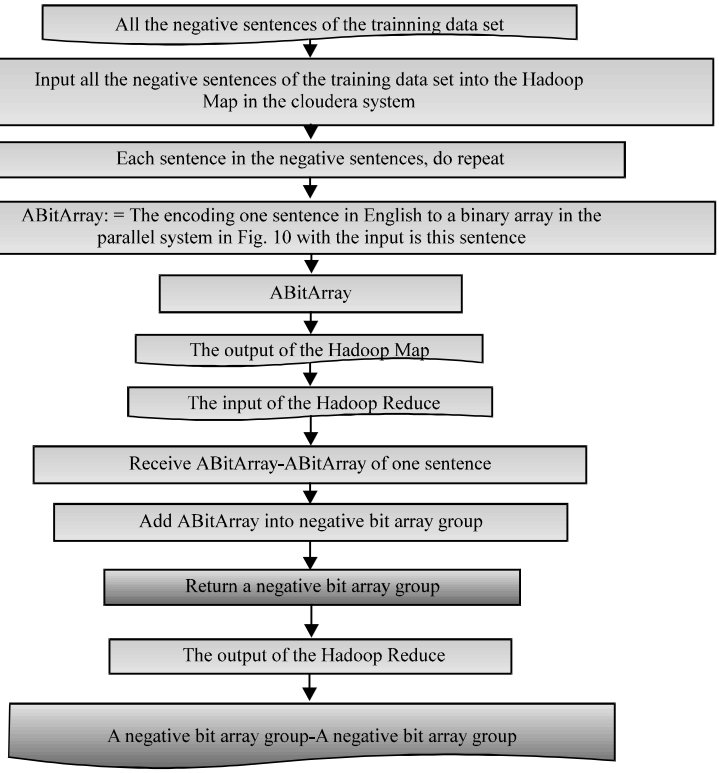


Fig. 12: Overview of encrypting all the negative sentences of the training data set to the bit arrays based on the bit arrays of the sentiment lexicons of the bESD in the distributed environment, called the negative bit array group

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 ANegativeBitArrayGroup

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 isthe 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 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. 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

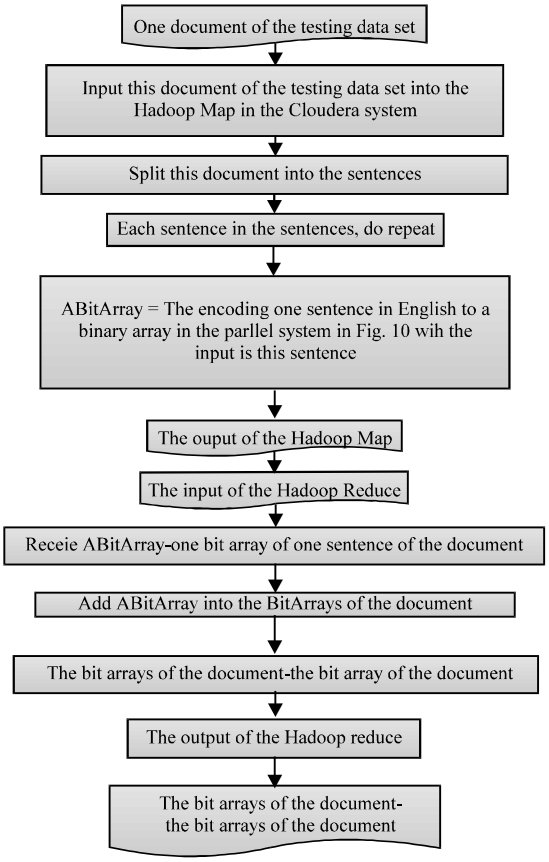


Fig. 13: Overview of transferring one document of the testing data set into the bit arrays of the document in the parallel system

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 isthe bit array clustered into either the positive bit array group or the negative bit array group of the training data set.

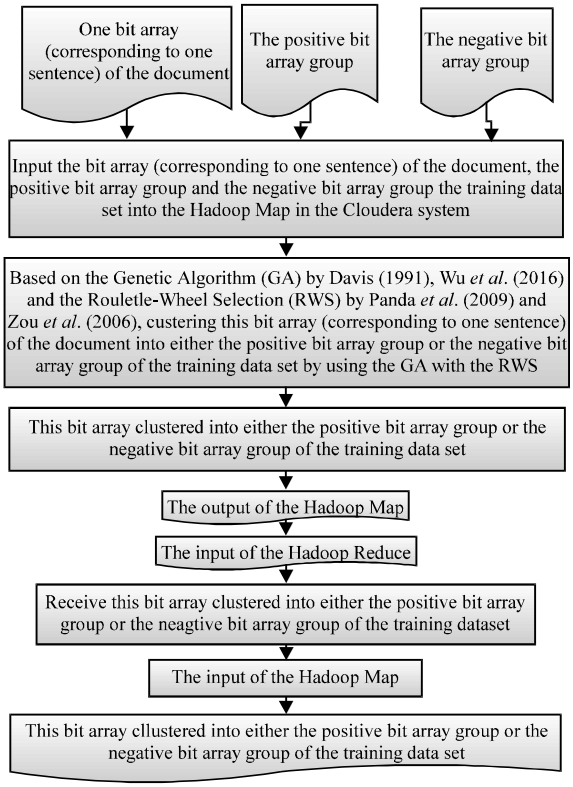


Fig. 14: Overview of clustering one bit array (corresponding to one sentence) of the document into either the positive bit array group or the negative bit array group of the training data set by using the GA with the RWS in the distributed system

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 thebit 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)

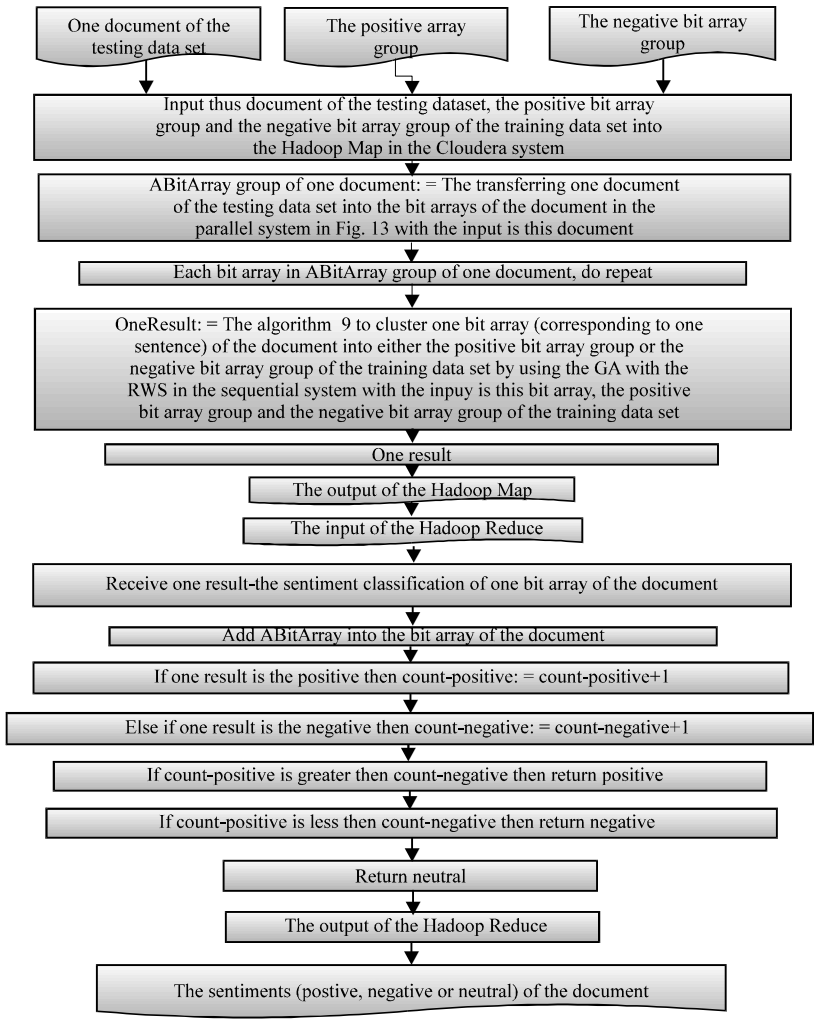


Fig. 15: Overview of clustering one document of the testing data set into either the positive bit array group or the negative bit array group of the training data set using the GA with the RWS in the distributed environment

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

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
Output: the sentiments (positive, negative or neutral)
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

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: OneResult: = 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 basis 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

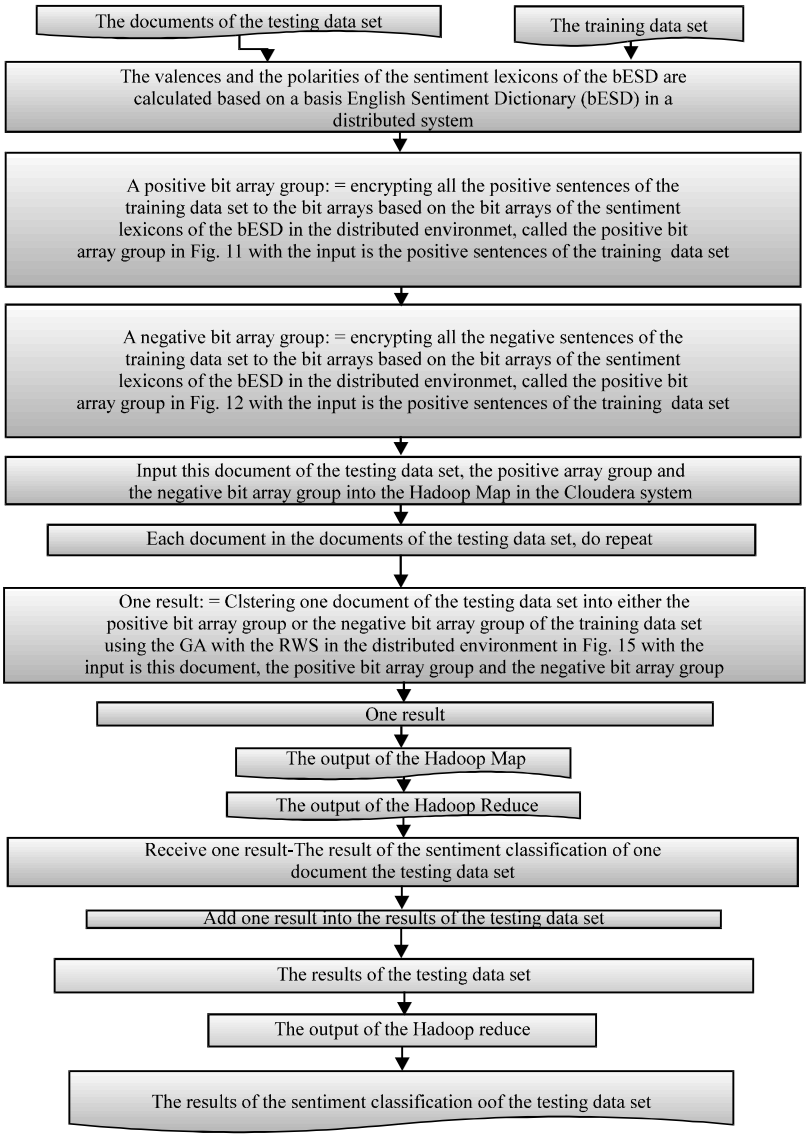


Fig. 16: Overview of performing the Hadoop Map phase of clustering the documents of the testing data set into either the positive or the negative in the distributed environment

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 CHChe, 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 Fuction (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 Fuction (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 CC3-10600 ECC 1333 MHz LP Unbuffered DIMMs. The operating system of each server in the 9 servers is: Cloudera. All 9 nodes have the same configuration information. The Java language is used in programming the application of the proposed model related to the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with theRoulette-Wheel Selection (RWS) the Fitness Fuction (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 theEnglish 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 7: The accuracy of our new model for the English documents in the testing data set

Proposed model/Class	Accuracy
Our new model	
Negative/Positive	88.02%

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

Variables	Average time of the classification /9,000.000 English documents (sec)
The Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Fuction (FF) which is the Roulette-Wheel Selection (RWS) in the sequential environment	38,106.889
The Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Fuction (FF) which is the Roulette-Wheel Selection (RWS) in the Cloudera distributed system with 3 nodes	11,368.963
The Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Fuction (FF) which is the Roulette-Wheel Selection (RWS) in the Cloudera distributed system with 6 nodes	6,484.481
The Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Fitness Fuction (FF) which is the Roulette-Wheel Selection (RWS) in the Cloudera distributed system with 9 nodes	4,256.321

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), Erkaya and Uzmay (2016) and Wu *et al.* (2016)

Studies	HC	CT	Sentiment classification	PNS	SD	DT	Language
Davis (1991)	No	No	No	No	Yes	No	EL
Kora and Krishna (2016)	No	No	Yes	No	Yes	No	EL
Yang *et al.* (2016)	No	No	Yes	No	Yes	Yes	EL
Erkaya and Uzmay (2016)	No	No	Yes	No	Yes	Yes	EL
Wu *et al.* (2016)	No	No	Yes	No	Yes	Yes	EL
Our research	Yes	Yes	Yes	Yes	Yes	Yes	EL

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), Erkaya and Uzmay (2016) and Wu *et al.* (2016)

Researches	Approach	Advantages	Disadvantages
Davis (1991)	Genetic algorithms	This study is meant to be a practical guide for practitioners, not, say 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 hiring Davis himself as a consultant, reading his research is probably the quickest and easiest way to get off the ground for a first real GA application	No mention
Kora and Krishna (2016)	Bundle block detection using differential evolution and Levenberg Marquardt neural network	The classification accuracy by the DE with LMNN was 99.1% for the detection of BBB. 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 BBB	No mention
Yang *et al.* (2016)	A hybrid approach based on stochastic competitive Hopfield neural network and efficient Genetic algorithm for frequency assignment problem	In this survey, the researcher's first propose five optimal strategies to build an efficient Genetic algorithm. Then the researcher's 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	No mention

Table 10: Continue

Researches	Approach	Advantages	Disadvantages
Erkaya and Uzmay (2016)	Balancing of planar mechanisms having imperfect joints using neural Network-Genetic Algorithm (NN-GA) approach	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. A neural network was used to model several characteristics of joint clearance. Kinematic and dynamic analyses were achieved using continuous contact mode 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	No mention
Wu *et al.* (2016)	Genetic algorithm trajectory plan optimization for EAMA: EAST articulated maintenance arm, fusion engineering and design	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	No mention

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

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)

Studies	HC	CT	Sentiment classification	PNS	SD	DT	Language
Panda *et al.* (2009)	No	No	No	No	Yes	No	EL
Lee *et al.* (1998)	No	No	Yes	No	Yes	No	EL
Tat and Tao (2003)	No	No	Yes	No	Yes	Yes	EL
Lipowski and Lipowska (2012)	No	No	Yes	No	Yes	Yes	EL
Zou *et al.* (2006)	No	No	Yes	No	Yes	Yes	EL
Our research	Yes	Yes	Yes	Yes	Yes	Yes	EL

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)

Researches	Approach	Advantages	Disadvantages
Panda *et al.* (2009)	PLANET: massively parallel learning of tree ensembles with MapReduce	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	No mention
Lee *et al.* (1998)	Digital filter design using Genetic algorithm	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 I/sub 2/ and I/sub/spl infin//(minimax) and their impact on the convergence rate and the optimal result	No mention
Tat and Tao (2003)	Using GIS and Genetic algorithm in highway alignment optimization	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	No mention
Lipowski and Lipowska (2012)	Roulette-Wheel selection via. stochastic acceptance	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(logN)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	No mention

Table 12: Continue

Researches	Approach	Advantages	Disadvantages
Zou *et al.* (2006)	Dynamic load balancing based on Roulette Wheel selection	The method is based on both static resource configuration and dynamic route selection. In static resource configuration stage, the optimization distribution of traffic trunk on parallel LSPs is obtained by offline optimization algorithm. In dynamic route selection stage, the LSP is selected by roulette wheel selection. The model and details of the algorithms are given	No mention

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

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-h), Ngoc *et al.* (2017), Phu *et al.* (2017) and Phu *et al.* (2017a-h)

Studies	HC	CT	Sentiment classification	PNS	SD	DT	Language
Agarwal and Mittal (2016a)	No	No	Yes	NM	Yes	Yes	Yes
Agarwal and Mittal (2016b)	No	No	Yes	NM	Yes	Yes	NM
Canuto *et al.* (2016)	No	No	Yes	NM	Yes	Yes	EL
Ahmed and Danti (2016)	No	No	Yes	NM	Yes	Yes	NM
Phu and Tuoi (2014)	No	No	Yes	No	No	No	EL
Ngoc *et al.* (2014)	No	No	Yes	No	No	No	EL
Our research	Yes	Yes	Yes	Yes	Yes	Yes	Yes

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)

Studies	Approach	Positives	Negatives
Agarwal and Mittal (2016a, b)	The machine learning approaches applied to sentiment analysis based applications	The main emphasis of this survey is to discuss the research involved in applying machine learning methods, mostly for sentiment classification at document level. Machine learning-based approaches 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	No mention
Agarwal and Mittal (2016b)	Semantic orientation-based approach for sentiment analysis	This approach initially mines sentiment-bearing terms from the unstructured text and further computes the polarity of the terms. Most of the sentiment-bearing terms are multi-word features unlike bag-of -words, e.g.,"good movie,""nicecinematography," "nice Actors," etc. Performance of semantic orientation-based approach has been limited in the literature due to inadequate coverage of multi-word features	No mention
Canuto *et al.* (2016)	Exploiting new sentiment-based meta-level features for effective Sentiment analysis	Experimentsperformed with a substantial number of datasets (nineteen) demonstrate that the effectivenessof the proposed sentiment-based meta-level features is notonly superior to the traditional bag-of-words representation (by up to 16%) but also is also superior in most cases to state-of-artmeta-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 resultsin all tested datasets in all scenarios	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 totheir related reviews
Ahmed and Danti (2016)	Rule-based machine learning algorithms	The proposed approach is tested by experimentingwith 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 TP-rate depicts higher efficiency rate and lower FP-rate Comparative experiments on various rule-based machine learning algorithms have been performed through a ten-fold cross validation training model for sentiment classification	No mention
Phu and Tuoi (2014)	The combination of term-counting method and enhanced contextual valence shifters method	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	No mention

Table 14: Continue:

Studies	Approach	Positives	Negatives
		researcher's combine five dictionaries into a new one with 21,137 entries. The new dictionary has many verbs, adverbs, phrases and idioms that were not in five dictionaries before. The study shows that the researcher's proposed method based on the combination of term-counting method and enhanced contextual valence shifters method has improved the accuracy of sentiment classification. The combined method has accuracy 68.984% on the testing dataset and 69.224% on the training dataset. All of these methods are implemented to classify the reviews based on our new dictionary and the internet movie database data set	
Ngoc *et al.* (2017)	Naive Bayes Model with N-gram method, negation handling method, Chi-square method and good-turing discounting, etc.	The researcher's have explored the Naive Bayes Model with N-gram method, negation handling method, Chi-square method and good-turing discounting by selecting different thresholds of good-turing discounting method and different minimum frequencies of Chi-square method to improve the accuracy of sentiment classification	No mention

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 usingthe Hamann Coefficient (HC) and the Genetic Algorithm (GA) with theRoulette-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 theRoulette-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 usingthe Hamann Coefficient (HC) and the Genetic Algorithm (GA) with theRoulette-Wheel Selection (RWS) 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 theRoulette-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 theRoulette-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 the Hamann Coefficient (HC) and the Genetic Algorithm (GA) with the Roulette-Wheel Selection (RWS) the Fitness Fuction (FF) to classify semantics of English documents based on sentences. The proposed model can process millions of documents in the shortest time. This study can be performed in distributed systems to shorten the execution time of the proposed model. It can be applied to other languages. It can save a lot of the storage spaces. 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 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 withthe 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, 2017a-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.* (2017a-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 besd 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 besd 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 basis 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 basis English Sentiment Dictionary (bESD)
Begin
Step 1: Receive this term
Step 2: Add this term into the basis 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-l), we get a bit array of this letter
Step 6: Add the bit array of this letter into ABitArray
Step 7: End Repeat-End Step 3
Step 8: Based on on the transferring a decimal to a binary code 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 besd 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 besd 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 A NegativeBitArrayGroup
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 (cα 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: Set count_positive : = 0 and count_negative: = 0
Step 3: Each bit array in AbitArray 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 OneResult is the positive Then count_positive: = count_positive+1
Step 6: Else If OneResult 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 TheResultsOfTheTestingDataSet: = null
Step 5: Each document in the documents of the testing data set, do repeat
Step 6: OneResult: = 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 OneResult into The Results Of The Testing DataSet
Step 8: End Repeat – End Step 5
Step 9: Return TheResultsOfTheTestingDataSet
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 (2017 g-l), 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 on the transferring a decimal to a binary code by Anonymous (2017 m-r), 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 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. 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 Abit Array 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:a 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 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:

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 bESDto perform the Hadoop Reduce phase in the parallel system, called the negative bit array group:

Input: ABitArray-a bit array of one sentence
Output:a 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 The BitArrays Of The Document
Begin
Step 1:Receive ABitArray
Step 2: Add ABitArray into The BitArrays Of The Document
Step 3: Return The BitArrays 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: OneResult 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

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. 16 with the input is this document
Step 3: Each bit array in AbitArray 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: Return OneResult;//the output of the Hadoop map
End

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 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
Begin
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
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 the testing data set the output of the Hadoop Map
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 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 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 := 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 results 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 Results of the Testing DataSet
Step 3: Return The Results of the testing DataSet

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

Agarwal, B. and N. Mittal, 2016. Machine Learning Approach for Sentiment Analysis. In: Prominent Feature Extraction for Sentiment Analysis, Agarwal, B. and N. Mittal (Eds.). Springer, Germany, ISBN:978-3-319-25341-1, pp: 21-45.

Agarwal, B. and N. Mittal, 2016. Machine Learning Approach for Sentiment Analysis. In: Prominent Feature Extraction for Sentiment Analysis, Agarwal, B. and N. Mittal (Eds.). Springer, Germany, ISBN:978-3-319-25341-1, pp: 21-45.

Ahmed, S. and A. Danti, 2016. Effective Sentimental Analysis and Opinion Mining of Web Reviews using Rule based Classifiers. In: Computational Intelligence in Data Mining, Behera, H. and D. Mohapatra (Eds.). Springer, New Delhi, India, ISBN:978-81-322-2732-8, pp: 171-179.

Anonymous, 2017a. 3.8. converting decimal numbers to binary numbers. RSI Ltd. http://interactivepython.org/runestone/static/pythonds/BasicDS/ConvertingDecimalNumberstoBinaryNumbers.html#converting-decimal-numbers-to-binary-numbers

Anonymous, 2017b. ASCII alphabet characters. American Standard Code for Information Interchange, USA. http://www.kerryr.net/pioneers/ascii2.htm.

Anonymous, 2017c. ASCII codes table. American Standard Code for Information Interchange, USA.

Anonymous, 2017d. ASCII codes table. American Standard Code for Information Interchange, USA.

Anonymous, 2017e. ASCII table, ASCII codes. American Standard Code for Information Interchange, USA. http://www.theasciicode.com.ar/.

Anonymous, 2017f. Binary to decimal conversion. AspenCore, Aspen, Colorado. http://www.electronics-tutorials.ws/binary/bin_2.html.

Anonymous, 2017g. Binary to decimal converter. RapidTables.com. https://www.rapidtables.com/convert/number/binary-to-decimal.html

Anonymous, 2017h. Converting from decimal to binary. Khan Academy, California, USA. https://www.khanacademy.org/math/algebra-home/alg-intro-to-algebra/algebra-alternate-number-bases/v/decimal-to-binary.

Anonymous, 2017i. Decimal to binary converter. Binary Hex Decimal Converter. https://www.binaryhexconverter.com/decimal-to-binary-converter

Anonymous, 2017j. How to convert from decimal to binary. WikiHow, Palo Alto, California, USA. https://www.wikihow.com/Convert-from-Decimal-to-Binary.

Anonymous, 2017k. The ASCII character set. American Standard Code for Information Interchange, USA. http://ee.hawaii.edu/~tep/EE160/Book/chap4/subsection2.1.1.1.html.

Bai, A., H. Hammer, A. Yazidi and P. Engelstad, 2014. Constructing sentiment lexicons in Norwegian from a large text corpus. Proceedings of the 2014 IEEE 17th International Conference on Computational Science and Engineering (CSE'14), December 19-21, 2014, IEEE, Chengdu, China, ISBN:978-1-4799-7980-6, pp: 231-237.

Brooke, J., M. Tofiloski and M. Taboada, 2009. Cross-linguistic sentiment analysis: From English to Spanish. Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'09), September 14-16, 2009, Association for Computational Linguistics, Borovets, Bulgaria, pp: 50-54.

Canuto, S., M.A. Goncalves and F. Benevenuto, 2016. Exploiting new sentiment-based meta-level features for effective sentiment analysis. Proceedings of the 9th ACM International Conference on Web Search and Data Mining (WSDM'16), February 22-25, 2016, ACM, San Francisco, California, USA., ISBN:978-1-4503-3716-8, pp: 53-62.

Choi, S.S., S.H. Cha and C.C. Tappert, 2010. A survey of binary similarity and distance measures. J. Syst. Cybern. Inf., 8: 43-48.

Dat, N.D., V.N. Phu, V.T.N. Tran, V.T.N. Chau and T.A. Nguyen, 2017. STING algorithm used English sentiment classification in a parallel environment. Intl. J. Pattern Recognit. Artif. Intell., 31: 1-30.

David, H.G., S.D. Hamann and R.B. Thomas, 1959. The second virial coefficients of some cyclic hydrocarbons. Aust. J. Chem., 12: 309-318.

Davis, L., 1991. Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, USA., ISBN:9780442001735, Pages: 385.

Du, W., S. Tan, X. Cheng and X. Yun, 2010. Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon. Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM'10), February 04-06, 2010, ACM, New York, USA., ISBN:978-1-60558-889-6, pp: 111-120.

Erkaya, S. and I. Uzmay, 2016. Balancing of Planar Mechanisms having Imperfect Joints using Neural Network-Genetic Algorithm (NN-GA) Approach. In: Dynamic Balancing of Mechanisms and Synthesizing of Parallel Robots, Zhang, D. and B. Wei (Eds.). Springer, Germany, ISBN:978-3-319-17682-6, pp: 299-317.

Feng, S., L. Zhang, B. Li, D. Wang and G. Yu *et al*., 2013. Is Twitter a better corpus for measuring sentiment similarity?. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, October 18-21, 2013, Association for Computational Linguistics, Seattle, Washington, USA., pp: 897-902.

Grigsby, J., K. Kaye, J. Baxter, S.M. Shetterly and R.F. Hamman, 1998. Executive cognitive abilities and functional status among community-dwelling older persons in the San Luis Valley health and aging study. J. Am. Geriatrics Soc., 46: 590-596.

Hernandez-Ugalde, J.A., J. Mora-Urpi and O.J. Rocha, 2011. Genetic relationships among wild and cultivated populations of peach palm (Bactris gasipaes Kunth, Palmae): Evidence for multiple independent domestication events. Genet. Resour. Crop Evol., 58: 571-583.

Htait, A., S. Fournier and P. Bellot, 2016. LSIS at SemEval-2016 task 7: Using web search engines for English and Arabic unsupervised sentiment intensity prediction. Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16), June 16-17, 2016, Association for Computational Linguistics, San Diego, California, pp: 469-473.

Ji, X., S.A. Chun, Z. Wei and J. Geller, 2015. Twitter sentiment classification for measuring public health concerns. Social Netw. Anal. Min., 5: 1-25.

Jiang, T., J. Jiang, Y. Dai and A. Li, 2015. Micro-blog emotion orientation analysis algorithm based on Tibetan and Chinese mixed text. Proceedings of the International Symposium on Social Science (ISSS'15), August 29-30, 2015, Atlantis Press, Atlanta, pp: 157-162.

Jovanoski, D., V. Pachovski and P. Nakov, 2015. Sentiment Analysis in Twitter for Macedonian. In: Proceedings of the International Conference Recent Advances in Natural Language Processing, Angelova, G., K. Bontcheva and R. Mitkov (Eds.). INCOMA Ltd., Hissar, Bulgaria, pp: 249-257.

Kora, P. and K.S.R. Krishna, 2016. Bundle Block Detection using Genetic Neural Network. In: Information Systems Design and Intelligent Applications, Satapathy, S., J. Mandal, S. Udgata and V. Bhateja (Eds.). Springer, New Delhi, ISBN:978-81-322-2750-2, pp: 309-317.

Lee, A., M. Ahmadi, G.A. Jullien, W.C. Miller and R.S. Lashkari, 1998. Digital filter design using genetic algorithm. Proceedings of the 1998 IEEE Symposium on Advances in Digital Filtering and Signal Processing, June 5-6, 1998, IEEE, Victoria, Canada, pp: 34-38.

Lipowski, A. and D. Lipowska, 2012. Roulette-wheel selection via stochastic acceptance. Phys. A Stat. Mech. Appl., 391: 2193-2196.

Malouf, R. and T. Mullen, 2017. Graph-based user classification for informal online political discourse. J. Comput. Sci., 1: 1-8.

Mao, H., P. Gao, Y.Y. Wang and J. Bollen, 2014. Automatic construction of financial semantic orientation lexicon from large-scale Chinese news corpus. Inst. Louis Bachelier, 20: 1-18.

Meyer, A.S., A.F. Garcia and A.P. Souza, 2004. Comparison of similarity coefficients used for cluster analysis with dominant markers in maize (*Zea mays* L.). Genet. Mol., 27: 83-91.

Mladenovic, S.D., A. Nikolic and V. Peric, 2008. Cluster analysis of soybean genotypes based on RAPD markers. Proceedings of the 43rd Croatian and 3rd International Symposium on Agriculture, February 18-21, 2008, University of Zagreb, Opatija, Croatia, pp: 367-370.

Netzer, O., R. Feldman, J. Goldenberg and M. Fresko, 2012. Mine your own business: Market-structure surveillance through text mining. Marketing Sci., 31: 521-543.

Ngoc, P.V., C.V.T. Ngoc, T.V.T. Ngoc and D.N. Duy, 2017. A C4.5 algorithm for English emotional classification. Evoling Syst., 1: 1-27.

Omar, N., M. Albared, A.A.Q. Shabi and A.T. Moslmi, 2013. Ensemble of classification algorithms for subjectivity and sentiment analysis of arabic customers' reviews. Int. J. Advancements Comput. Technol., 5: 77-85.

Panda, B., J.S. Herbach, S. Basu and R.J. Bayardo, 2009. Planet: massively parallel learning of tree ensembles with mapreduce. Proc. VLDB. Endowment, 2: 1426-1437.

Phu, V.N. and P.T. Tuoi, 2014. Sentiment classification using enhanced contextual valence shifters. Proceedings of the 2014 International Conference on Asian Language Processing (IALP'14), October 20-22, 2014, IEEE, Kuching, Malaysia, ISBN:978-1-4799-5331-8, pp: 224-229.

Phu, V.N., N.D. Dat, V.T.N. Tran, V.T.N. Chau and T.A. Nguyen, 2017g. Fuzzy C-means for english sentiment classification in a distributed system. Appl. Intell., 46: 717-738.

Phu, V.N., V.T.N. Chau and V.T.N. Tran, 2017d. SVM for English semantic classification in parallel environment. Intl. J. Speech Technol., 20: 487-508.

Phu, V.N., V.T.N. Chau and V.T.N. Tran, 2017e. Shifting semantic values of English phrases for classification. Intl. J. Speech Technol., 20: 509-553.

Phu, V.N., V.T.N. Chau, N.D. Dat, V.T.N. Tran and T.A. Nguyen, 2017. A valences-totaling model for English sentiment classification. Knowl. Inf. Syst., 53: 579-636.

Phu, V.N., V.T.N. Chau, V.T.N. Tran and N.D. Dat, 2017c. A Vietnamese adjective emotion dictionary based on exploitation of Vietnamese language characteristics. Artif. Intell. Rev., 1: 1-67.

Phu, V.N., V.T.N. Chau, V.T.N. Tran, D.N. Duy and K.L.D. Duy, 2017a. A valence-totaling model for Vietnamese sentiment classification. Evolving Syst., 1: 1-47.

Phu, V.N., V.T.N. Tran, V.T.N. Chau, D.N. Duy and K.L.D. Duy, 2017b. Semantic lexicons of English nouns for classification. Evolving Syst., 1: 1-65.

Phu, V.N., V.T.N. Tran, V.T.N. Chau, N.D. Dat and K.L.D. Duy, 2017f. A decision tree using ID3 algorithm for English semantic analysis. Intl. J. Speech Technol., 20: 593-613.

Ponomarenko, J.V., P.E. Bourne and I.N. Shindyalov, 2002. Building an automated classification of DNA-binding protein domains. Bioinf., 18: S192-S201.

Ren, Y., N. Kaji, N. Yoshinaga and M. Kitsuregawa, 2014. Sentiment classification in under-resourced languages using graph-based semi-supervised learning methods. IEICE. Trans. Inf. Syst., 97: 790-797.

Ren, Y., N. Kaji, N. Yoshinaga, M. Toyoda and M. Kitsuregawa, 2011. Sentiment classification in resource-scarce languages by using label propagation. Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation (PACLIC'11), December 16-18, 2011, Nanyang Technological University, Singapore, ISBN:978-4-905166-02-3, pp: 420-429.

Scheible, C., 2010. Sentiment translation through lexicon induction. Proceedings of the 2010 Workshop on ACL Student Research (ACLstudent'10), July 13, 2010, Association for Computational Linguistics, Uppsala, Sweden, pp: 25-30.

Schnyer, D.M., M. Verfaellie, M.P. Alexander, G. LaFleche and L. Nicholls *et al*., 2004. A role for right medial prefrontal cortex in accurate feeling-of-knowing judgments: Evidence from patients with lesions to frontal cortex. Neuropsychologia, 42: 957-966.

Schraw, G., 1995. Measures of feeling-of-knowing accuracy: A new look at an old problem. Appl. Cognit. Psychol., 9: 321-332.

Shikalgar, N.R. and A.M. Dixit, 2014. JIBCA: Jaccard Index Based Clustering Algorithm for mining online review. Intl. J. Comput. Appl., 105: 23-28.

Tamas, J., J. Podani and P. Csontos, 2001. An extension of presence/absence coefficients to abundance data: A new look at absence. J. Veg. Sci., 12: 401-410.

Tan, S. and J. Zhang, 2008. An empirical study of sentiment analysis for Chinese documents. Expert Syst. Appl., 34: 2622-2629.

Tat, C.W. and F. Tao, 2003. Using GIS and Genetic algorithm in highway alignment optimization. Proceedings of the 2003 IEEE conference on Intelligent Transportation Systems Vol. 2, October 12-15, 2003, IEEE, Shanghai, China, pp: 1563-1567.

Turney, P.D. and M.L. Littman, 2002. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Comput. Sci., 1: 1-13.

Urbina, E.M., R.P. Wadwa, C. Davis, B.M. Snively and L.M. Dolan *et al*, 2010. 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 J. Pediatr., 156: 731-737.

Wan, X., 2009. Co-training for cross-lingual sentiment classification. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP Vol. 1, August 02-07, 2009, Association for Computational Linguistics, Suntec, Singapore, ISBN:978-1-932432-45-9, pp: 235-243.

Wang, G. and K. Araki, 2007. Modifying SO-PMI for Japanese weblog opinion mining by using a balancing factor and detecting neutral expressions. Proceedings of the Conference of the North American Chapter of the Association on Computational Linguistics Human Language Technologies; Companion Volume, Short Papers (NAACL-Short'07), April 22-27, 2007, Association for Computational Linguistics, Rochester, New York, USA., pp: 189-192.

Wu, J., H. Wu, Y. Song, Y. Cheng and W. Zhao *et al*., 2016. Genetic algorithm trajectory plan optimization for EAMA: EAST articulated maintenance ARM. Fusion Eng. Des., 109: 700-706.

Yang, G., S. Wu, Q. Jin and J. Xu, 2016. A hybrid approach based on stochastic competitive hopfield neural network and efficient genetic algorithm for frequency assignment problem. Appl. Soft Comput., 39: 104-116.

Zhang, Z., Q. Ye, W. Zheng and Y. Li, 2010. Sentiment classification for consumer word-of-mouth in Chinese: Comparison between supervised and unsupervised approaches. Proceedings of the 2010 International Conference on E-Business Intelligence (ICEBI'10), December 19-21, 2010, Atlantis Press, Atlanta, ISBN:978-90-78677-40-6, pp: 427-433.

Zou, Y., Z. Mi and M. Xu, 2006. Dynamic load balancing based on roulette wheel selection. Proceedings 2006 International Conference on Communications, Circuits and Systems Vol. 3, June 25-28, 2006, IEEE, Guilin, China, pp: 1732-1734.