Journal of Theoretical and Applied Information Technology
15th June 2018. Vol.96. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A SELF-TRAINING - BASED MODEL USING A K-NN ALGORITHM AND THE SENTIMENT LEXICONS - BASED MULTI-DIMENSIONAL VECTORS OF A S6 COEFFICIENT FOR SENTIMENT CLASSIFICATION

[1]DR.VO NGOC PHU, [2]DR.VO THI NGOC TRAN

[1]Nguyen Tat Thanh University, 300A Nguyen Tat Thanh Street, Ward 13, District 4, Ho Chi Minh City, 702000, Vietnam

[2]School of Industrial Management (SIM), Ho Chi Minh City University of Technology - HCMUT, Vietnam National University, Ho Chi Minh City, Vietnam

E-mail: [1]vongocphu03hca@gmail.com, vongocphu@ntt.edu.vn, [2]vtntran@HCMUT.edu.vn

ABSTRACT

Many surveys and commercial applications of sentiment classification have already applied to many different fields in everyday life, such as in political activities, commodity production, and commercial activities significantly. A semi-supervised learning of a machine learning used for a new model for big data sentiment classification has already been built in this survey. We have proposed a novel model using mainly a self-training (ST) approach to classify 12,500,000 documents of our testing data set comprising the 6,250,000 positive and the 6,250,000 negative into 2,000 documents of our training data set including the 1,000 positive and the 1,000 negative in English. In this self-training model (STM), a K-Nearest Neighbors algorithm (K-NN) has been used in training a classifier according to many multi-dimensional vectors of sentiment lexicons of a S6 coefficient (S6C). After training this classifier of the STM of each loop, the 100 documents of the testing data set have certainly been chosen, and then, they have been added to this classifier. The sentiment classification of all the documents of the testing data set has been identified after many loops of training the classifier of the STM certainly. In this survey, we do not use any vector space modeling (VSM). We do not use any one-dimensional vectors according to both the VSM and the sentiment classification. The S6C is used in creating the sentiment classification of our basis English sentiment dictionary (bESD) through a Google search engine with AND operator and OR operator. The novel model has firstly been performed in a sequential system and then, we have secondly implemented the proposed model in a parallel network environment. The results of the sequential environment are less than that in the distributed system. We have achieved 89.13% accuracy of the testing data set. The results of the proposed model can widely be used in many commercial applications and surveys of the sentiment classification.

Keywords: *English sentiment classification; parallel system; Cloudera; Hadoop Map and Hadoop Reduce; Balanced Iterative Reducing and Clustering using Hierarchies; K-NN; S6 coefficient.*

1. INTRODUCTION

This guide provides

Many surveys and commercial applications of sentiment classification have already applied to many different fields in everyday life, such as in political activities, commodity production, and commercial activities significantly.

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

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Sentiment classification is done by using machine-learning methods or methods based on lexicons; or a combination of both. Opinion mining relates to emotional researches that are expressed in documents. Sentiment classification has a wide range of applications in the fields of business, organizations, governments and individuals. In recent studies, a common approach is classifying text messages into categories such as positive, negative, or neutral support for a subject matter. Sentiment dictionaries are usually used for looking up the sentiments of individual words. There are the surveys related the sentiment lexicons in [1-32].

In this work, the studies related to a S6 coefficient (S6C) are in [39, 44].

There are the researches related to a K-Nearest Neighbors algorithm (K-NN) in [45-49] as follows: K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique. KNN is a non-parametric lazy learning algorithm. That is a pretty concise statement. When you say a technique is non-parametric, it means that it does not make any assumptions on the underlying data distribution. This is pretty useful, as in the real world, most of the practical data does not obey the typical theoretical assumptions made (eg gaussian mixtures, linearly separable etc.). Non parametric algorithms like KNN come to the rescue here, etc.

The studies related to the Self-Training algorithm are in [50-54] as follows: Semi-supervised learning is to learn a hypothesis by combining information in both labeled and unlabeled data. Self-training is a well-known semi-supervised algorithm. In self-training process, a base learner is firstly trained onlabeled set. Then, iteratively, it attempts to choose to label several examples that it is most confident of in the unlabeled set, etc.

The basic principles are proposed as follows:

1)We assume that each English sentence has m English words (or English phrases).

2)We assume that the maximum number of one English sentence is m_max; it means that m is less than m_max or m is equal to m_max.

3)We assume that each English document has n English sentences.

4)We assume that the maximum number of one English document is n_max; it means that n is less than n_max or n is equal to n_max.

The motivation of this new model is as follows: A S6 coefficient (S6C) can be applied to a distributed network environment. Cosine, Ochiai, Sorensen, Tanimoto, PMI and Jaccard measures are used popularly to calculate the emotional values of the words. Thus, other similar measures can be used to identify the semantic scores of the words. Many algorithms in the data mining field can be applied to natural language processing, specifically semantic classification for processing millions of English documents. A K-Nearest Neighbors algorithm (K-NN) can be applied to the sentiment classification. A self-training (ST) algorithm can use the K-NN to classify the sentiments (positive, negative, or neutral) for the documents of the testing data set. This will result in many discoveries in scientific research, hence the motivation for this study.

The novelty and originality of the proposed approach are as follows:

1)The S6C was applied to the sentiment classification.

2)This algorithm can also be applied to identify the emotions of millions of documents. The K-NN was applied to the sentiment classification.

3)The ST was applied to the sentiment classification.

4)The ST used the K-NN as a classifier for training.

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

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

7)We did not use the VSM [55-57].

8)We used many sentiment lexicons.

9)We did not use any one-dimensional vectors according to both the VSM and the sentiment lexicons.

10)We used many multi-dimensional vectors based on the sentiment lexicons.

11)We used the S6C through a Google search engine with AND operator and OR operator to identify the sentiment values and polarities of the sentiment lexicons in English.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

12)The input of this survey is the documents of the testing data set and the documents of the training data set in English. We studied to transfer the documents into the formats for the novel model which can process them.

13)We tested the proposed model in both a sequential environment and a distributed network system.

14)We built the S6C – related equations and the K-NN – related algorithms in this survey.

15)We proposed the ST – related algorithms in both a sequential system and a parallel network environment.

Therefore, we will study this model in more detail.

According to the purpose of this work, we always try to find a new approach to improve many accuracies of the results of the sentiment classification and to shorten many execution times of the proposed model with a low cost.

We want to get higher accuracy of the results of the sentiment classification and shorten execution time of the sentiment classification, we do not use any VSM. We use the sentiment lexicons. We use any one-dimensional vectors according to both the VSM and the sentiment classification. The S6C is used in creating the sentiment classification of our basis English sentiment dictionary (bESD) through a Google search engine with AND operator and OR operator. This novel model uses the ST. The ST uses the K-NN as a classifier. The sentiment lexicons are used in transferring one document into one multi-dimensional vector.

The documents of our training data set are transferred into the multi-dimensional vectors which are the input of the K-NN of the ST model.

The novel model is implemented as follows: we firstly calculate the valences of the sentiment lexicons of the bESD by using the S6C through the Google search engine with AND operator and OR operator. One document is transferred into one multi-dimensional vector. We transfer the positive documents of the training data set into the positive multi-dimensional vectors, called the positive group. We also transfer the negative documents of the training data set into the negative multi-dimensional vectors, called the negative group. The self-training (ST) approach is mainly used for the novel model. In this self-training model (STM), a K-Nearest Neighbors algorithm (K-NN) has been used in training a classifier according to the multi-dimensional vectors. The documents of the testing data set are transferred into the multi-dimensional vectors. The input of the K-NN is the positive group and the negative group of the training data set; the multi-dimensional vectors of the testing data set. After training this classifier of the STM of each loop, the 100 multi-dimensional vectors of the testing data set have certainly been chosen, and then, they have been added to this classifier. The multi-dimensional vectors of the 100 multi-dimensional vectors of the testing data set clustered into the positive group are added to the positive group of the classifier. The multi-dimensional vectors of the 100 multi-dimensional vectors of the testing data set clustered into the negative group are added to the negative group of the classifier. The sentiment classification of all the documents of the testing data set has been identified after many loops of training the classifier of the STM certainly.

In the sequential environment, all the above things are performed to get an accuracy of the result of the sentiment classification and an execution time of the result of the sentiment classification of the proposed model. Then, in the parallel network environment, all the above things are secondly implemented to shorten the execution times of the proposed model to get the accuracy of the results of the sentiment classification and the execution times of the results of the sentiment classification of our novel model.

The significant contributions of the novel model can be applied to many areas of research as well as commercial applications as follows:

1)Many surveys and commercial applications can use the results of this work in a significant way.

2)The algorithms are built in the proposed model.

3)This survey can certainly be applied to other languages easily.

4)The results of this study can significantly be applied to the types of other words in English.

5)Many crucial contributions are listed in the Future Work section.

6)The algorithm of data mining is applicable to semantic analysis of natural language processing.

7)This study also proves that different fields of scientific research can be related in many ways.

8)Millions of English documents are successfully processed for emotional analysis.

9)The sentiment classification is implemented in the parallel network environment.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

10)The principles are proposed in the research.

11)The Cloudera distributed environment is used in this study.

12)The proposed work can be applied to other distributed systems.

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

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

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

16)The S6C – related equations are proposed in this survey.

17)The K-NN – related algorithms are built in this study.

18)The ST – related algorithms are proposed in this work.

This study contains 6 sections. Section 1 introduces the study; Section 2 discusses the related works about the S6 coefficient (S6C), K-NN algorithm, self-training (ST) algorithm, etc.; Section 3 is about the English data set; Section 4 represents the methodology of our proposed model; Section 5 represents the experiment. Section 6 provides the conclusion. The References section comprises all the reference documents; all tables are shown in the Appendices section.

2. RELATED WORK

We summarize many researches which are related to our research.

By far, we know that PMI (Pointwise Mutual Information) equation and SO (Sentiment Orientation) equation are used for determining polarity of one word (or one phrase), and strength of sentiment orientation of this word (or this phrase). Jaccard measure (JM) is also used for calculating polarity of one word and the equations from this Jaccard measure are also used for calculating strength of sentiment orientation this word in other research. PMI, Jaccard, Cosine, Ochiai, Tanimoto, and Sorensen measure are the similarity measure between two words; from those, we prove that the S6 coefficient (S6C) is also used for identifying valence and polarity of one English word (or one English phrase). Finally, we identify the sentimental values of English verb phrases based on the basis English semantic lexicons of the basis English emotional dictionary (bESD).

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

Two studies related to the PMI measure and Jaccard measure are in [14, 15]. In the survey [14], the authors empirically evaluate the performance of different corpora in sentiment similarity measurement, which is the fundamental task for word polarity classification. The research in [15] proposes a new method to estimate impression of short sentences considering adjectives. In the proposed system, first, an input sentence is 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 works related to the Jaccard measure are in [16-22]. The survey in [16] investigates the problem of sentiment analysis of the online review. In the study [17], the authors are addressing the issue of spreading public concern about epidemics. Public concern about a communicable disease can be seen as a problem of its own, etc.

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

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

There are the works related to the S6 coefficient (S6C) in [39-44]. The authors in [39] collected 76 binary similarity and distance measures used over the last century and reveal their correlations through the hierarchical clustering technique. In [40], this was a prospective observational case series. A total of 60 normal eyes of 60 subjects included in the study. SD-OCT macular scanning (macular cube 512×128 scan) was performed twice by an experienced examiner. The average retinal thicknesses of the nine macular sectors as defined

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

by the Early Treatment Diabetic Retinopathy Study (ETDRS) were recorded. Each coefficient of repeatability was calculated for the macular thickness measurements of the ETDRS subfields, etc.

3. DATA SET

In Figure 1, we built our the testing data set including the 12,500,000 documents in the movie field, which contains the 6,250,000 positive and 6,250,000 negative in English. All the documents in our English testing data set are automatically extracted from English Facebook, English websites and social networks; then we labeled positive and negative for them.

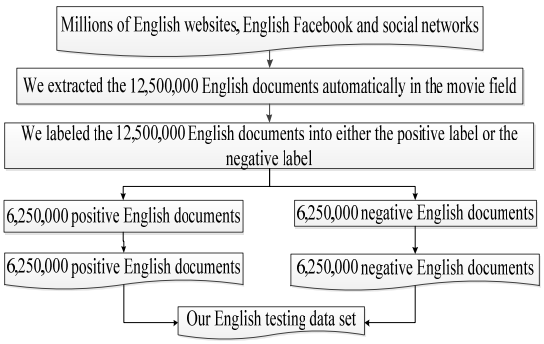

Figure 1: Our testing data set in English.

In Figure 2 below, we built our training data set including the 2,000 documents in the movie field, which contains the 1,000 positive and 1,000 negative in English. All the documents in our English training data set are automatically extracted from English Facebook, English websites and social networks; then we labeled positive and negative for them.

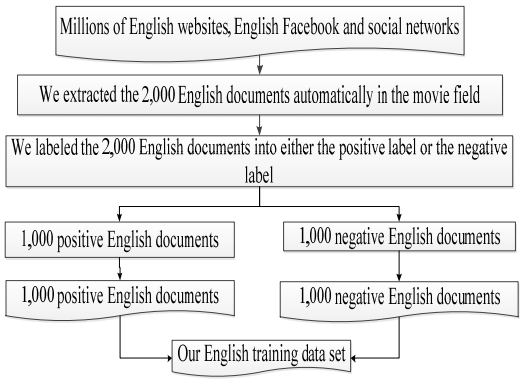

Figure 2: Our training data set in English.

4. METHODOLOGY

We implement the proposed model in Figure 3. This section comprises three parts. The first part is the sub-section (4.1) which we create the sentiment lexicons in English in both a sequential environment and a distributed system. The second part is the sub-section (4.2) which we transfer all the documents of the testing data set and the training data set into the multi-dimensional vectors in a sequential system and a parallel network environment. The third part is the sub-section (4.3) which we use the self-training model using the K-NN algorithm and the S6C to classify 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.

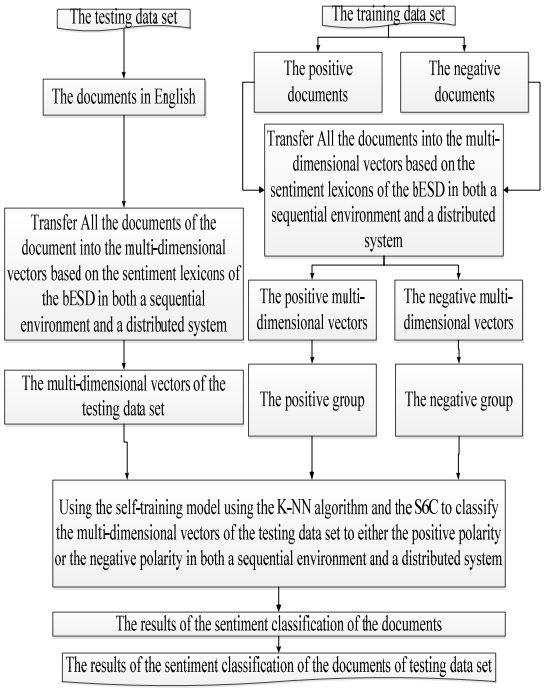

Figure 3: Overview of our novel model.

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

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

sentiment dictionary (bESD) in a parallel environment.
In the sub-section (4.2), the section comprises three sub-sections as follows: (4.2.1) and (4.2.2). In the sub-section (4.2.1), we transfer all the documents of the testing data set and the training data set into the multi-dimensional vectors in a sequential system. In the sub-section (4.2.2), we transfer all the documents of the testing data set and the training data set into the multi-dimensional vectors in a parallel network environment.
In the sub-section (4.3), the section comprises two parts as follows: (4.3.1) and (4.3.2). In the sub-section (4.3.1), we use the self-training model using the K-NN algorithm and the S6C 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 sub-section (4.3.2), we use the self-training model using the K-NN algorithm and the S6C to classify the documents of the testing data set into either the positive vector group or the negative vector group in a distributed system.

4.1 Creating the sentiment lexicons in English

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

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

In this part, we calculate the valence and the polarity of one English word (or phrase) by using the S6C through a Google search engine with AND operator and OR operator, as the following diagram in Figure 3 below shows.

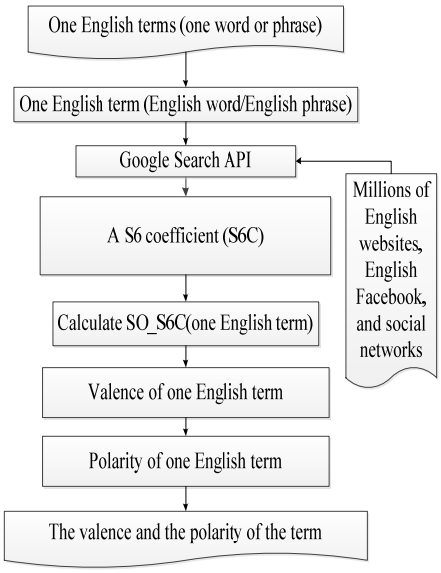

Figure 3: Overview of identifying the valence and the polarity of one term in English using a S6 coefficient (S6C)

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

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

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

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

In [1-8] the positive and the negative of Eq. (2) in English are: positive = {good, nice, excellent, positive, fortunate, correct, superior} and negative = {bad, nasty, poor, negative, unfortunate, wrong, inferior}.
The AltaVista search engine is used in the PMI equations of [2, 3, 5] and the Google search engine is used in the PMI equations of [4, 6, 8]. Besides, [4] also uses German, [5] also uses Macedonian, [6] also uses Arabic, [7] also uses Chinese, and [8] also uses Spanish. In addition, the Bing search engine is also used in [6].
With [9-12], the PMI equations are used in Chinese, not English, and Tibetan is also added in [9]. About the search engine, the AltaVista search engine is used in [11] and [12] and uses three search engines, such as the Google search engine, the Yahoo search engine and the Baidu search engine. The PMI equations are also used in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Japanese with the Google search engine in [13]. [14] and [15] also use the PMI equations and Jaccard equations with the Google search engine in English.

The Jaccard equations with the Google search engine in English are used in [14, 15, 17]. [16] and [21] use the Jaccard equations in English. [20] and [22] use the Jaccard equations in Chinese. [18] uses the Jaccard equations in Arabic. The Jaccard equations with the Chinese search engine in Chinese are used in [19].

The authors in [28] used the Ochiai Measure through the Google search engine with AND operator and OR operator to calculate the sentiment values of the words in Vietnamese. The authors in [29] used the Cosine Measure through the Google search engine with AND operator and OR operator to identify the sentiment scores of the words in English. The authors in [30] used the Sorensen Coefficient through the Google search engine with AND operator and OR operator to calculate the sentiment values of the words in English. The authors in [31] used the Jaccard Measure through the Google search engine with AND operator and OR operator to calculate the sentiment values of the words in Vietnamese. The authors in [32] used the Tanimoto Coefficient through the Google search engine with AND operator and OR operator to identify the sentiment scores of the words in English

With the above proofs, we have 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 Vietnamse. The Ochiai is used with the Google in Vietnamese. The Cosine and Sorensen are used with the Google in English.

According to [1-32], PMI, Jaccard, Cosine, Ochiai, Sorensen, Tanimoto and S6 coefficient (S6C) are the similarity measures between two words, and they can perform the same functions and with the same characteristics; so S6C is used in calculating the valence of the words. In addition, we prove that S6C can be used in identifying the valence of the English word through the Google search with the AND operator and OR operator.

With the S6 coefficient (S6C) in [39-44], we have the equation of the S6C:

$$\text{S6 Coefficient}(a,b) = \text{S6 Measure}(a,b) = \text{S6C}(a,b) = \frac{(a \cap b)}{\sqrt{[(a \cap b) + (\neg a \cap b)] * [(a \cap b) + (a \cap \neg b)]}} * \frac{(\neg a \cap \neg b)}{\sqrt{[(\neg a \cap b) + (\neg a \cap \neg b)] * [(a \cap \neg b) + (\neg a \cap \neg b)]}} \quad (3)$$

with a and b are the vectors.

From the eq. (1), (2), (3), we propose many new equations of the S6C to calculate the valence and the polarity of the English words (or the English phrases) through the Google search engine as the following equations below.

In eq. (3), when a has only one element, a is a word. When b has only one element, b is a word. In eq. (3), a is replaced by w1 and b is replaced by w2.

$$\text{S6 Measure}(w1,w2) = \text{S6 Coefficient}(w1,w2) = \text{S6C}(w1,w2) = \frac{P(w1,w2)}{\sqrt{[P(w1,w2) + P(\neg w1,w2)] * [P(w1,w2) + P(w1,\neg w2)]}} * \frac{P(\neg w1,\neg w2)}{\sqrt{[P(\neg w1,w2) + P(\neg w1,\neg w2)] * [P(w1,\neg w2) + P(\neg w1,\neg w2)]}} \quad (4)$$

Eq. (3) is similar to eq. (1). In eq. (2), eq. (1) is replaced by eq. (4). We have eq. (5) as follows:

$$\text{Valence}(w) = \text{SO\_S6C}(w) = \text{S6C}(w, \text{positive\_query}) - \text{S6C}(w, \text{negative\_query}) \quad (5)$$

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

$$\text{S6C}(w, \text{positive\_query}) = \frac{P(w, \text{positive\_query})}{\sqrt{[P(w, \text{positive\_query}) + P(\neg w, \text{positive\_query})] * [P(w, \text{positive\_query}) + P(w, \neg \text{positive\_query})]}} * \frac{P(\neg w, \neg \text{positive\_query})}{\sqrt{[P(\neg w, \text{positive\_query}) + P(\neg w, \neg \text{positive\_query})] * [P(w, \neg \text{positive\_query}) + P(\neg w, \neg \text{positive\_query})]}} \quad (6)$$

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

$$\text{S6C}(w, \text{negative\_query}) = \frac{P(w, \text{negative\_query})}{\sqrt{[P(w, \text{negative\_query}) + P(\neg w, \text{negative\_query})] * [P(w, \text{negative\_query}) + P(w, \neg \text{negative\_query})]}} * \frac{P(\neg w, \neg \text{negative\_query})}{\sqrt{[P(\neg w, \text{negative\_query}) + P(\neg w, \neg \text{negative\_query})] * [P(w, \neg \text{negative\_query}) + P(\neg w, \neg \text{negative\_query})]}} \quad (7)$$

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

1)w, w1, w2 : are the English words (or the English phrases)

2)P(w1, w2): number of returned results in Google search by keyword (w1 and w2). We use the Google Search API to get the number of returned results in search online Google by keyword (w1 and w2).

3)P(w1): number of returned results in Google search by keyword w1. We use the Google Search API to get the number of returned results in search

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

online Google by keyword w1.
4)P(w2): number of returned results in Google search by keyword w2. We use the Google Search API to get the number of returned results in search online Google by keyword w2.
5)Valence(W) = SO_S6C(w): valence of English word (or English phrase) w; is SO of word (or phrase) by using the S6 coefficient (S6C)
6)positive_query: { active or good or positive or beautiful or strong or nice or excellent or fortunate or correct or superior } with the positive query is the a group of the positive English words.
7)negative_query: { passive or bad or negative or ugly or week or nasty or poor or unfortunate or wrong or inferior }
with the negative_query is the a group of the negative English words.
8)P(w, positive_query): number of returned results in Google search by keyword (positive_query and w). We use the Google Search API to get the number of returned results in search online Google by keyword (positive_query and w)
9)P(w, negative_query): number of returned results in Google search by keyword (negative_query and w). We use the Google Search API to get the number of returned results in search online Google by keyword (negative_query and w)
10)P(w): number of returned results in Google search by keyword w. We use the Google Search API to get the number of returned results in search online Google by keyword w
11)P(¬w,positive_query): number of returned results in Google search by keyword ((not w) and positive_query). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and positive_query).
12)P(w, ¬positive_query): number of returned results in the Google search by keyword (w and (not (positive_query))). We use the Google Search API to get the number of returned results in search online Google by keyword (w and [not (positive_query)]).
13)P(¬w, ¬positive_query): number of returned results in the Google search by keyword (w and (not (positive_query))). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and [not (positive_query)]).
14)P(¬w,negative_query): number of returned results in Google search by keyword ((not w) and negative_query). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and negative_query).
15)P(w,¬negative_query): number of returned results in the Google search by keyword (w and (not (negative_query))). We use the Google Search API to get the number of returned results in search online Google by keyword (w and (not (negative_query))).
16)P(¬w,¬negative_query): number of returned results in the Google search by keyword (w and (not (negative_query))). We use the Google Search API to get the number of returned results in search online Google by keyword ((not w) and (not (negative_query))).

As like Cosine, Ochiai, Sorensen, Tanimoto, PMI and Jaccard about calculating the valence (score) of the word, we identify the valence (score) of the English word w based on both the proximity of positive_query with w and the remote of positive_query with w; and the proximity of negative_query with w and the remote of negative_query with w.
The English word w is the nearest of positive_query if S6C (w, positive_query) is as equal as 1.
The English word w is the farthest of positive_query if S6C(w, positive_query) is as equal as 0.
The English word w belongs to positive_query being the positive group of the English words if S6C(w, positive_query) > 0 and S6C(w, positive_query) ≤ 1.
The English word w is the nearest of negative_query if S6C(w, negative_query) is as equal as 1.
The English word w is the farthest of negative_query if S6C(w, negative_query) is as equal as 0.
The English word w belongs to negative_query being the negative group of the English words if S6C(w, negative_query) > 0 and S6C(w, negative_query) ≤ 1.
So, the valence of the English word w is the value of S6C(w, positive_query) substracting the value of S6C(w, negative_query) and the eq. (7) is the equation of identifying the valence of the English word w.
We have the information S6C as follows:
1)S6C(w, positive_query) ≥ 0 and S6C(w, positive_query) ≤ 1.
2)S6C(w, negative_query) ≥ 0 and S6C (w, negative_query) ≤ 1
3)If S6C (w, positive_query) = 0 and S6C (w, negative_query) = 0 then SO_S6C (w) = 0.
4)If S6C (w, positive_query) = 1 and S6C (w, negative_query) = 0 then SO_S6C (w) = 0.
5)If S6C (w, positive_query) = 0 and S6C (w, negative_query) = 1 then SO_S6C (w) = -1.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

6)If S6C (w, positive_query) = 1 and S6C (w, negative_query) = 1 then SO_S6C(w) = 0.
So, SO_S6C (w) ≥ -1 and SO_S6C (w) ≤ 1.
The polarity of the English word w is positive polarity If SO_S6C (w) > 0. The polarity of the English word w is negative polarity if SO_S6C (w) < 0. The polarity of the English word w is neutral polarity if SO_S6C (w) = 0. In addition, the semantic value of the English word w is SO_S6C (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. (5), eq. (6), and eq. (7), we identify the valence and the polarity of one word (or one phrase) in English by using the S6C through the Google search engine with AND operator and OR operator.
In Table 1, we present the comparisons of our model's results with the works related to [1-32].
The comparisons of our model's advantages and disadvantages with the works related to [1-32] are shown in Table 2.
In Table 3, we display the comparisons of our model's results with the works related to the S6 coefficient (S6C) in [39, 44].
The comparisons of our model's benefits and drawbacks with the studies related to the S6 coefficient (S6C) in [39, 44] are presented in Table 4.

4.1.2 Creating a basis English sentiment dictionary (bESD) in a sequential environment

According to [33-38], we have at least 55,000 English terms, including nouns, verbs, adjectives, etc. In this part, we calculate the valence and the polarity of the English words or phrases for our basis English sentiment dictionary (bESD) by using the S6C in a sequential system, as the following diagram in Figure 4 below shows.

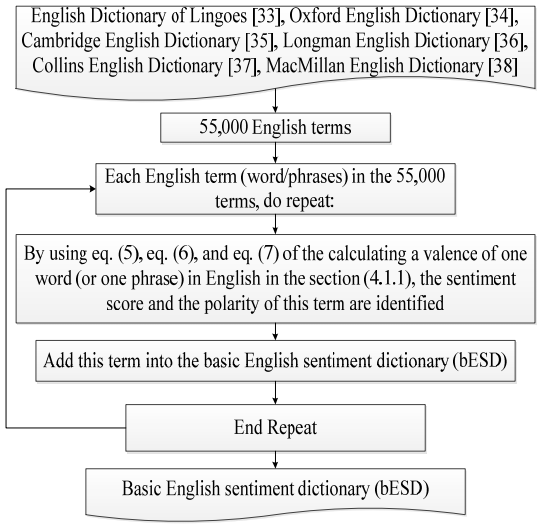

Figure 4: Overview of creating a basis English sentiment dictionary (bESD) in a sequential environment

We proposed the algorithm 1 to perform this section. The main ideas of the algorithm 1 are as follows:
Input: the 55,000 English terms; the Google search engine
Output: a basis English sentiment dictionary (bESD)
Step 1: Each term in the 55,000 terms, do repeat:
Step 2: By using eq. (5), eq. (6), and eq. (7) of the calculating a valence of one word (or one phrase) in English in the section (4.1.1), the sentiment score and the polarity of this term are identified. The valence and the polarity are calculated by using the S6C through the Google search engine with AND operator and OR operator.
Step 3: Add this term into the basis English sentiment dictionary (bESD);
Step 4: End Repeat – End Step 1;
Step 5: Return bESD;
Our basis English sentiment dictionary (bESD) has more 55,000 English words (or English phrases) and bESD is stored in Microsoft SQL Server 2008 R2.

4.1.3 Creating a basis English sentiment dictionary (bESD) in a distributed system

According to [33-38], we have at least 55,000 English terms, including nouns, verbs, adjectives, etc. In this part, we calculate the valence and the polarity of the English words or phrases for our basis English sentiment dictionary (bESD) by using the S6C in a parallel network environment, as the following diagram in Figure 5 below shows.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

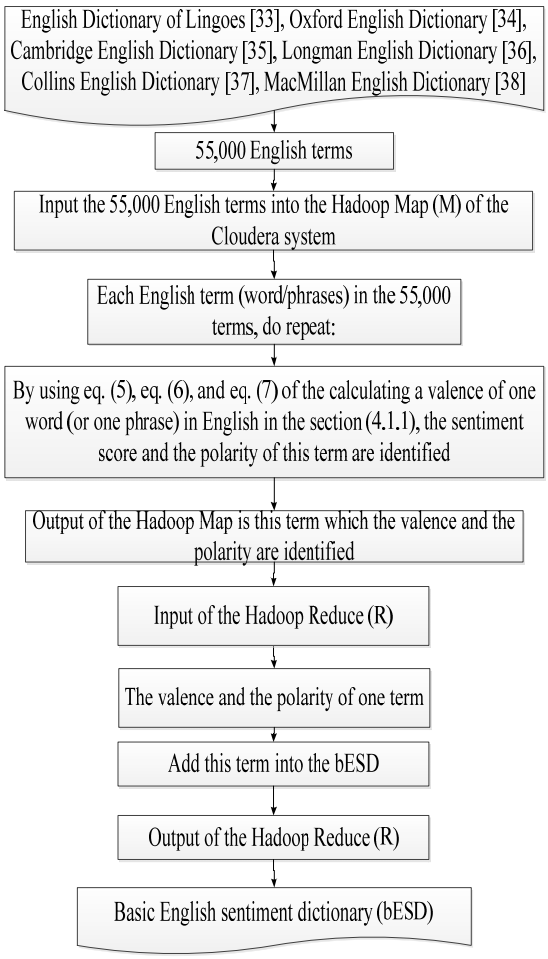

Figure 5: Overview of creating a basis English sentiment dictionary (bESD) in a distributed environment

In Figure 5, this section includes two phases: the Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase is the 55,000 terms in English in [33-38]. The output of the Hadoop Map phase is one term which the sentiment score and the polarity are identified. The output of the Hadoop Map phase is the input of the Hadoop Reduce phase. Thus, the input of the Hadoop Reduce phase is one term which the sentiment score and the polarity are identified. The output of the Hadoop Reduce phase is the basis English sentiment dictionary (bESD).

We proposed the algorithm 2 to implement the Hadoop Map phase. The main ideas of the algorithm 2 are as follows:

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

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

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

Step 2: By using eq. (5), eq. (6), and eq. (7) of the calculating a valence of one word (or one phrase) in English in the section (4.1.1), the sentiment score and the polarity of this term are identified. The valence and the polarity are calculated by using the S6C through the Google search engine with AND operator and OR operator.

Step 3: Return this term;

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

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

Output: a basis English sentiment dictionary (bESD)

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

Step 2: Return bESD;

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

4.2 Transferring all the documents of the testing data set and the training data set into the multi-dimensional vectors in a sequential system and a parallel network environment.

In this section, we transfer all the documents of the testing data set and the training data set into the multi-dimensional vectors in a sequential system and a parallel network environment.

The section comprises three sub-sections as follows: (4.2.1) and (4.2.2). In the sub-section (4.2.1), we transfer all the documents of the testing data set and the training data set into the multi-dimensional vectors in a sequential system. In the sub-section (4.2.2), we transfer all the documents of the testing data set and the training data set into the multi-dimensional vectors in a parallel network environment.

4.2.1 Transferring all the documents of the testing data set and the training data set into the multi-dimensional vectors in a sequential system

In this section, we transfer all the documents of the testing data set and the training data set into the multi-dimensional vectors in a sequential system.

We build the algorithm 4 to transfer one English document into one multi-dimensional vector according to the sentiment lexicons of the bESD in the sequential environment. The main ideas of the algorithm 4 are as follows:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Input: one English document
Output: the multi-dimensional vector
Step 1: Split the English document into many separate sentences based on "." Or "!" or "?";
Step 2: Set Multi-dimensionalVector := { } { } with n_max rows and m_max columns;
Step 3: Set i := 0;
Step 4: Each sentence in the sentences of this document, do repeat:
Step 5: Multi-dimensionalVector[i][] := {};
Step 6: Set j := 0;
Step 7: Split this sentence into the meaningful terms (meaningful words or meaningful phrases);
Step 8: Get the valence of this term based on the sentiment lexicons of the bESD;
Step 9: Add this term into Multi-dimensionalVector[i];
Step 10: Set j := j+ 1;
Step 11: End Repeat – End Step 4;
Step 12: While j is less than m_max, repeat:
Step 13: Add {0} into Multi-dimensionalVector[i];
Step 14: Set j := j+1;
Step 15: End Repeat – End Step 12;
Step 16: Set i := i+1;
Step 17: End Repeat – End Step 4;
Step 18: While i is less than n_max, repeat:
Step 19: Add the vector {0} into Multi-dimensionalVector;
Step 20: Set i := i+1;
Step 21: End Repeat – End Step 18;
Step 22: Return Multi-dimensionalVector;

We propose the algorithm 5 to transfer all the documents of the testing data set into the multi-dimensional vectors based on the sentiment lexicons of the bESD in the sequential environment. The main ideas of the algorithm 5 are as follows:
Input: the documents of the testing data set
Output: the multi-dimensional vectors of the testing data set
Step 1: Set TheMulti-dimensionalVectors := {}
Step 2: Each document in the documents of the testing data set, do repeat:
Step 3: OneMulti-dimensionalVector := the algorithm 4 to transfer one English document into one multi-dimensional vector according to the sentiment lexicons of the bESD in the sequential environment with the input is this document;
Step 4: Add OneMulti-dimensionalVector into TheMulti-dimensionalVectors;
Step 5: End Repeat- End Step 2;
Step 6: Return TheMulti-dimensionalVectors;

We build the algorithm 6 to transfer all the positive documents of the training data set into all the multi-dimensional vectors based on the sentiment lexicons of the bESD, called the positive group of the training data set in the sequential system. The main ideas of the algorithm 6 are as follows:
Input: all the positive documents of the training data set;
Output: the positive multi-dimensional vectors, called the positive group - ThePositiveMulti-dimensionalVectors
Step 1: Set ThePositiveMulti-dimensionalVectors := null;
Step 2: Each document in the positive documents, repeat:
Step 3: Multi-dimensionalVector := the algorithm 4 to transfer one English document into one multi-dimensional vector according to the sentiment lexicons of the bESD in the sequential environment with the input is this document;
Step 4: Add Multi-dimensionalVector into ThePositiveMulti-dimensionalVectors;
Step 5: End Repeat – End Step 2;
Step 6: Return ThePositiveMulti-dimensionalVectors;

We implement the algorithm 7 to transfer all the negative sentences of the training data set into all the one-dimensional vectors based on the sentiment lexicons of the bESD, called the negative group of the training data set in the sequential environment. The main ideas of the algorithm 7 are as follows:
Input: all the negative sentences of the training data set;
Output the negative multi-dimensional vectors, called the negative vector group - TheNegativeMulti-dimensionalVectors;
Step 1: Set TheNegativeMulti-dimensionalVectors := null;
Step 2: Each document in the negative documents, repeat:
Step 3: Multi-dimensionalVector := the algorithm 4 to transfer one English document into one multi-dimensional vector according to the sentiment lexicons of the bESD in the sequential environment with the input is this document;
Step 4: Add Multi-dimensionalVector into TheNegativeMulti-dimensionalVectors;
Step 5: End Repeat – End Step 2;
Step 6: Return TheNegativeMulti-dimensionalVectors;

4.2.2 Transferring all the documents of the testing data set and the training data set into the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

multi-dimensional vectors in a parallel network environment

In this section, we transfer all the documents of the testing data set and the training data set into the multi-dimensional vectors in a parallel system.

In Figure 6, we transfer one English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera. This stage includes two phases: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map phase is one sentence and the bESD. The output of the Hadoop Map phase is one term (one meaningful word/or one meaningful phrase) which the valence is identified. The input of the Hadoop Reduce phase is the output of the Hadoop Map, thus, the input of the Hadoop Reduce phase is one term (one meaningful word/or one meaningful phrase) which the valence is identified. The output of the Hadoop Reduce phase is one one-dimensional vector of this sentence.

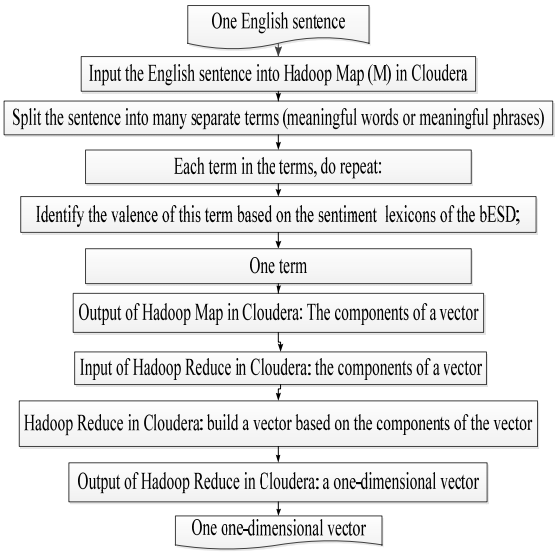

Figure 6: Overview of transforming each English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera

We build the algorithm 8 to perform the Hadoop Map phase of transferring each English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera. The main ideas of the algorithm 8 are as follows:
Input: one sentence and the bESD;
Output: one term (one meaningful word/or one meaningful phrase) which the valence is identified
Step 1: Input this sentence and the bESD into the Hadoop Map in the Cloudera system;
Step 2: Split this sentence into the many meaningful terms (meaningful words/or meaningful phrases) based on the bESD;
Step 3: Each term in the terms, do repeat:
Step 4: Identify the valence of this term based on the bESD;
Step 5: Return this term; //the output of the Hadoop Map phase.

We propose the algorithm 9 to perform the Hadoop Reduce phase of transferring each English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera. The main ideas of the algorithm 9 are as follows:
Input: one term (one meaningful word/or one meaningful phrase) which the valence is identified – the output of the Hadoop Map phase
Output: one one-dimensional vector based on the sentiment lexicons of the bESD
Step 1: Receive one term;
Step 2: Add this term into the one-dimentional vector;
Step 3: Return the one-dimentional vector;

In Figure 7, we transfer one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system.

In Figure 7, this stage includes two phases: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map phase is one document. The output of the Hadoop Map is one one-dimensional vector. The input of the Hadoop Reduce is the Hadoop Map, thus, the input of the Hadoop Reduce is one one-dimensional vector. The output of the Hadoop Reduce is the multi-dimensional vector of this document.

We propose the algorithm 10 to implement the Hadoop Map phase of the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system. The main ideas of the algorithm 10 are as follows:
Input: one document
Output: one one-dimensional vector;//the output of the Hadoop Map in the Cloudera system
Step 1: Input this document into the Hadoop Map in the Cloudera system.
Step 2: Split this document into the sentences;
Step 3: Each sentence in the sentences, do repeat:
Step 4: One-dimensionalVector := null;
Step 5: Split this sentence into the meaningful terms;
Step 6: Each term in the meaningful terms, repeat:
Step 7: Get the valence of this term based on the sentiment lexicons of the bESD;
Step 8: Add this term into One-dimensionalVector;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 9 End Repeat – End Step 6;
Step 10: Return this One-dimensionalVector;
Step 11: The output of the Hadoop Map is this One-dimensionalVector;
We propose the algorithm 11 to implement the Hadoop Reduce phase of transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system. The main ideas of the algorithm 11 are as follows:
Input: One-dimensionalVector - one one-dimensional vector of the Hadoop Map (the input of the Hadoop Reduce is the output of the Hadoop Map)
Output: the multi-dimensional vector of the English document – Multi-dimensionalVector;
Step 1: Receive One-dimensionalVector;
Step 2: Add this One-dimensionalVector into One-dimensionalVector;
Step 3: Return Multi-dimensionalVector;

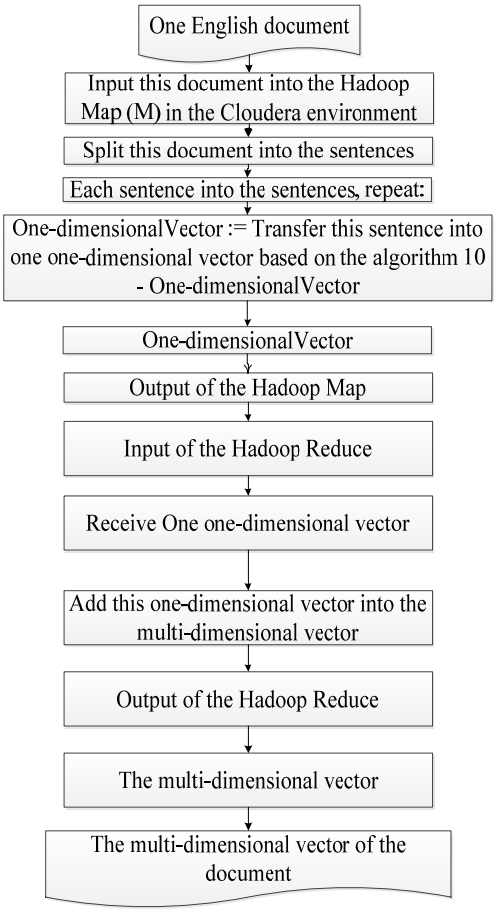

Figure 7: Overview of transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system

In Figure 8, we transfer the documents of the testing data set into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system as follows:

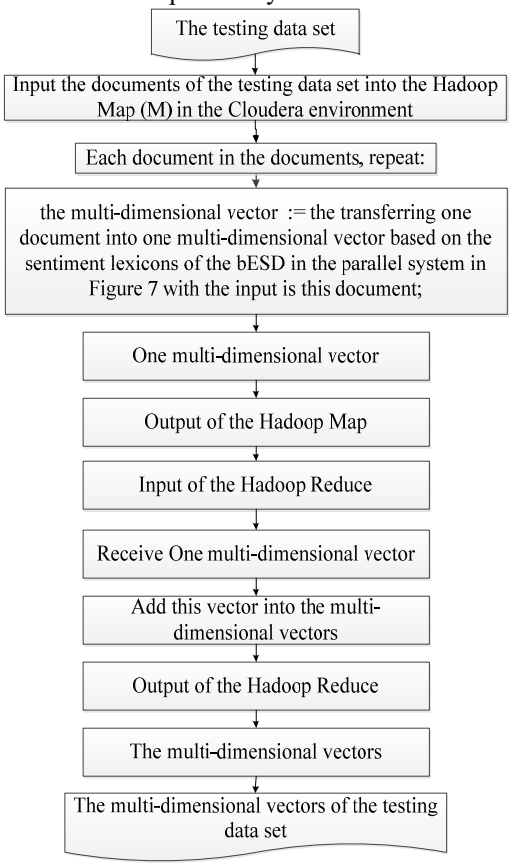

Figure 8: Overview of transferring the documents of the testing data set into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system

In Figure 8, this stage includes two phases: the Hadoop Map phase and the Hadoop Reduce phase. The input of the Hadoop Map phase is the documents of the testing data set. The output of the Hadoop Mp is one multi-dimensional vector. The input of the Hadoop Reduce is the Hadoop Map, thus, the input of the Hadoop Reduce is one multi-dimensional vector. The output of the Hadoop Reduce is the multi-dimensional vectors of the testing data set
We propose the algorithm 12 to implement the Hadoop Map phase of transferring one document into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system. The main ideas of the algorithm 12 are as follows:
Input: the documents of the testing data set

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Output: one multi-dimensional vector (corresponding to one document)
Step 1: Input the documents of the testing data set into the Hadoop Map in the Cloudera system.
Step 3: Each document in the documents of the testing data set, do repeat:
Step 4: the multi-dimensional vector := the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system in Figure 7 with the input is this document;
Step 5: Return this multi-dimensional vector;
Step 6: The output of the Hadoop Map is this multi-dimensional vector;
We propose the algorithm 13 to implement the Hadoop Reduce phase of transferring one document into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system. The main ideas of the algorithm 13 are as follows:
Input: one multi-dimensional vector of the Hadoop Map (the input of the Hadoop Reduce is the output of the Hadoop Map)
Output: the multi-dimensional vectors of the English documents of the testing data set
Step 1: Receive one multi-dimensional vector of the Hadoop Map
Step 2: Add this multi-dimensional vector into the multi-dimensional vectors of the testing data set;
Step 3: Return the multi-dimensional vectors of the testing data set;

In Figure 9, we transfer the positive documents of the training data set into the positive multi-dimensional vectors (called the positive group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

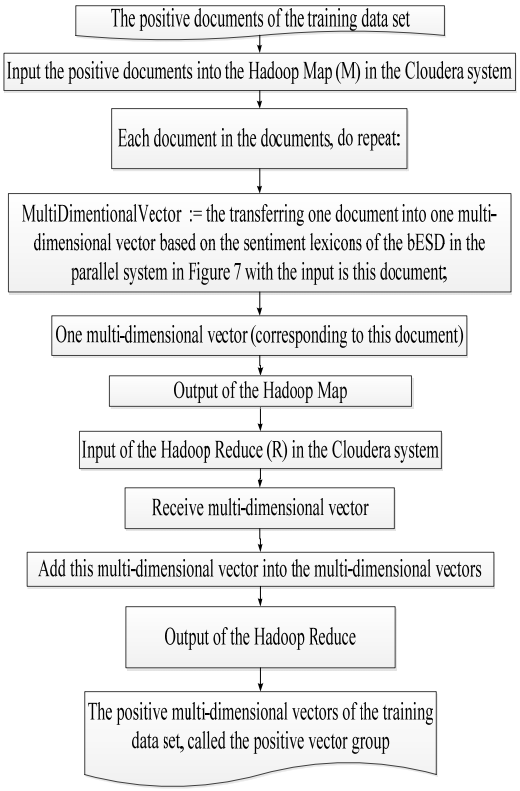

Figure 9: Overview of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

In Figure 9, the stage includes two phases: the Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase is the positive documents of the training data set. The output of the Hadoop Map phase is one multi-dimensional vector (corresponding to one document of the positive documents of the training data set). The input of the Hadoop Reduce phase is the output of the Hadoop Map phase, thus, the input of the Hadoop Reduce phase is one multi-dimensional vector (corresponding to one document of the positive documents of the training data set). The output of the Hadoop Reduce phase is the positive multi-dimensional vectors, called the positive group (corresponding to the positive documents of the training data set)
We propose the algorithm 14 to perform the Hadoop Map phase of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive group of the training data set) based on the sentiment lexicons of the bESD in the distributed

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

system. The main ideas of the algorithm 14 are as follows:
Input: the positive documents of the training data set
Output: one multi-dimensional vector (corresponding to one document of the positive documents of the training data set)
Step 1: Input the positive documents into the Hadoop Map in the Cloudera system.
Step 2: Each document in the documents, do repeat:
Step 3: MultiDimentionalVector := the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system in Figure 7 with the input is this document;
Step 4: Return MultiDimentionalVector ;
We propose the algorithm 15 to implement the Hadoop Reduce phase of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive vector group of the training data set) based on the sentiment lexicons of the bESD in the distributed system. The main ideas of the algorithm 15 are as follows:
Input: one multi-dimensional vector (corresponding to one document of the positive documents of the training data set)
Output: the positive multi-dimensional vectors, called the positive group (corresponding to the positive documents of the training data set)
Step 1: Receive one multi-dimensional vector;
Step 2: Add this multi-dimensional vector into PositiveVectorGroup;
Step 3: Return PositiveVectorGroup - the positive multi-dimensional vectors, called the positive group (corresponding to the positive documents of the training data set);

In Figure 10, we transfer the negative documents of the training data set into the negative multi-dimensional vectors (called the negative group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

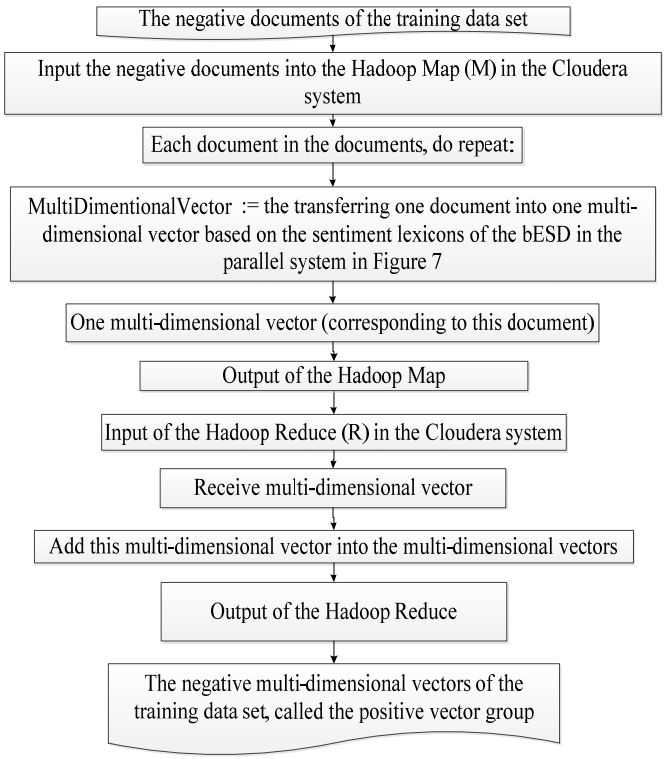

Figure 10: Overview of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

In Figure 10, the stage includes two phases: the Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase is the negative documents of the training data set. The output of the Hadoop Map phase is one multi-dimensional vector (corresponding to one document of the negative documents of the training data set). The input of the Hadoop Reduce phase is the output of the Hadoop Map phase, thus, the input of the Hadoop Reduce phase is one multi-dimensional vector (corresponding to one document of the negative documents of the training data set). The output of the Hadoop Reduce phase is the negative multi-dimensional vectors, called the negative group (corresponding to the negative documents of the training data set)
We build the algorithm 16 to perform the Hadoop Map phase of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative group of the training data set) based on the sentiment lexicons of the bESD in the distributed system. The main ideas of the algorithm 16 are as follows:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Input: the negative documents of the training data set
Output: one multi-dimensional vector (corresponding to one document of the negative documents of the training data set)
Step 1: Input the negative documents into the Hadoop Map in the Cloudera system.
Step 2: Each document in the documents, do repeat:
Step 3: MultiDimentionalVector := the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system in Figure 7
Step 4: Return MultiDimentionalVector ;
We propose the algorithm 17 to implement the Hadoop Reduce phase of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative group of the training data set) based on the sentiment lexicons of the bESD in the distributed system. The main ideas of the algorithm 17 are as follows:

Input: one multi-dimensional vector (corresponding to one document of the negative documents of the training data set)

Output: the negative multi-dimensional vectors, called the negative group (corresponding to the negative documents of the training data set)

Step 1: Receive one multi-dimensional vector;

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

Step 3: Return NegativeVectorGroup - the megative multi-dimensional vectors, called the negative group (corresponding to the negative documents of the training data set);

4.3 Using the self-training model using the K-NN algorithm and the S6C to classify 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.

In section, we use the self-training model using the K-NN algorithm and the S6C to classify 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 section comprises two parts as follows: (4.3.1) and (4.3.2). In the sub-section (4.3.1), we use the self-training model using the K-NN algorithm and the S6C 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 sub-section (4.3.2), we use the self-training model using the K-NN algorithm and the S6C to classify the documents of the testing data set into either the positive vector group or the negative vector group in a distributed system.

4.3.1 Using the self-training model using the K-NN algorithm and the S6C 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 this section, we use the self-training model using the K-NN algorithm and the S6C 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 Figure 11 as follows: According to the creating a basis English sentiment dictionary (bESD) in a sequential environment (4.1.2), we firstly calculate the valences of the sentiment lexicons of the bESD by using the S6C through the Google search engine with AND operator and OR operator. One document is transferred into one multi-dimensional vector. We transfer the positive documents of the training data set into the positive multi-dimensional vectors, called the positive group. We also transfer the negative documents of the training data set into the negative multi-dimensional vectors, called the negative group. The self-training (ST) approach is mainly used for the novel model. In this self-training model (STM), a K-Nearest Neighbors algorithm (K-NN) has been used in training a classifier according to the multi-dimensional vectors. The documents of the testing data set are transferred into the multi-dimensional vectors. The input of the K-NN is the positive group and the negative group of the training data set; the multi-dimensional vectors of the testing data set. After training this classifier of the STM of each loop, the 100 multi-dimensional vectors of the testing data set have certainly been chosen, and then, they have been added to this classifier. The multi-dimensional vectors of the 100 multi-dimensional vectors of the testing data set clustered into the positive group are added to the positive group of the classifier. The multi-dimensional vectors of the 100 multi-dimensional vectors of the testing data set clustered into the negative group are added to the negative group of the classifier. The sentiment classification of all the documents of the testing data set has been identified after many loops of training the classifier of the STM certainly.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

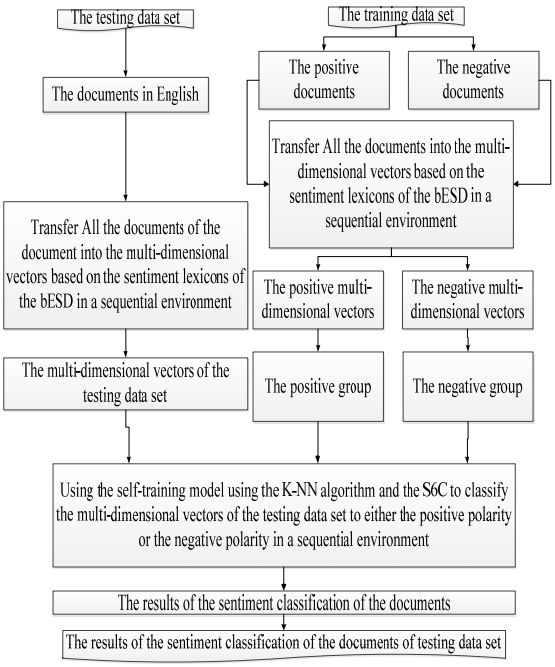

Figure 11: Overview of our novel model.

According to the researches related to a K-Nearest Neighbors algorithm (K-NN) in [45-49], we present the K-NN algorithm which is enhanced to be able to classify the sentiments (positive, negative, or neutral) for the documents. The main ideas of the K-NN are as follows:

1. Identify the K parameter (K - Nearest Neighbors): in this survey, we choose K = 2;
2. Calculate the distance between the vector (which need to be clustered) with the vectors in the training data by Euclidean distance;
3. Arrange the distances by acending order; and Identify K - nearest neighbors with the vectors which need to be clustered;
4. Get all clusters of K - nearest neighbors which are identified;
5. Based on all clusters of nearest neighbors to identify the cluster of the vector;

The K-NN uses Euclidean distance to calculate the distance between two vectors

We build the algorithm 18 to classify all the documents of the testing data set into either the positive or the negative in the sequential system by using the self-training model with the K-NN algorithm and the S6C. The main ideas of the algorithm 18 are as follows:

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

Output: the results of the sentiment classification of the documents of the testing data set (positive, negative, or neutral);

Step 1: the creating a basis English sentiment dictionary (bESD) in a sequential environment (4.1.2);

Step 2: the algorithm 6 to transfer all the positive documents of the training data set into all the multi-dimensional vectors based on the sentiment lexicons of the bESD, called the positive group of the training data set in the sequential system

Step 3: the algorithm 7 to transfer all the negative documents of the training data set into all the multi-dimensional vectors based on the sentiment lexicons of the bESD, called the negative group of the training data set in the sequential environment.

Step 4: the algorithm 5 to transfer all the documents of the testing data set into the multi-dimensional vectors based on the sentiment lexicons of the bESD in the sequential environment.

Step 5: L : a data set of the labeled documents of the testing data set as follows: the positive group and the negative group of the training data set

Step 6: U : a set of the un-labeled documents of the training data set as follows: the multi-dimensional vectors of the testing data set;

Step 7: While until U = ∅

Step 7.1: Begin: Training a classifier h on L and using h to classify the data in U: Using the K-NN with the input of the K-NN is the multi-dimensional vectors of the testing data set, the positive group and the negative group of the training data set;

Step 7.2: Choose the 100 multi-dimensional vectors of the testing data set which are the best results of Step 7.1.

Step 7.3: U := U - the 100 multi-dimensional vectors of the testing data set;

Step 7.4: Add the 100 multi-dimensional vectors of the testing data set into either the positive group or the negative group of the training data set as follows: L := L + the 100 multi-dimensional vectors of the testing data set

Step 7.5: End While – End Step 7;

Step 8: Return the results of the sentiment classification of the documents of the testing data set (positive, negative, or neutral);

4.3.2 Using the self-training model using the K-NN algorithm and the S6C to classify the documents of the testing data set into either the positive vector group or the negative vector group in a distributed system.

In this section, we use the self-training model using the K-NN algorithm and the S6C to classify the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

documents of the testing data set into either the positive vector group or the negative vector group in a distributed network environment in Figure 12 as follows: Based on the creating a basis English sentiment dictionary (bESD) in a distributed system (4.1.3), we firstly calculate the valences of the sentiment lexicons of the bESD by using the S6C through the Google search engine with AND operator and OR operator. One document is transferred into one multi-dimensional vector. We transfer the positive documents of the training data set into the positive multi-dimensional vectors, called the positive group. We also transfer the negative documents of the training data set into the negative multi-dimensional vectors, called the negative group. The self-training (ST) approach is mainly used for the novel model. In this self-training model (STM), a K-Nearest Neighbors algorithm (K-NN) has been used in training a classifier according to the multi-dimensional vectors. The documents of the testing data set are transferred into the multi-dimensional vectors. The input of the K-NN is the positive group and the negative group of the training data set; the multi-dimensional vectors of the testing data set. After training this classifier of the STM of each loop, the 100 multi-dimensional vectors of the testing data set have certainly been chosen, and then, they have been added to this classifier. The multi-dimensional vectors of the 100 multi-dimensional vectors of the testing data set clustered into the positive group are added to the positive group of the classifier. The multi-dimensional vectors of the 100 multi-dimensional vectors of the testing data set clustered into the negative group are added to the negative group of the classifier. The sentiment classification of all the documents of the testing data set has been identified after many loops of training the classifier of the STM certainly.

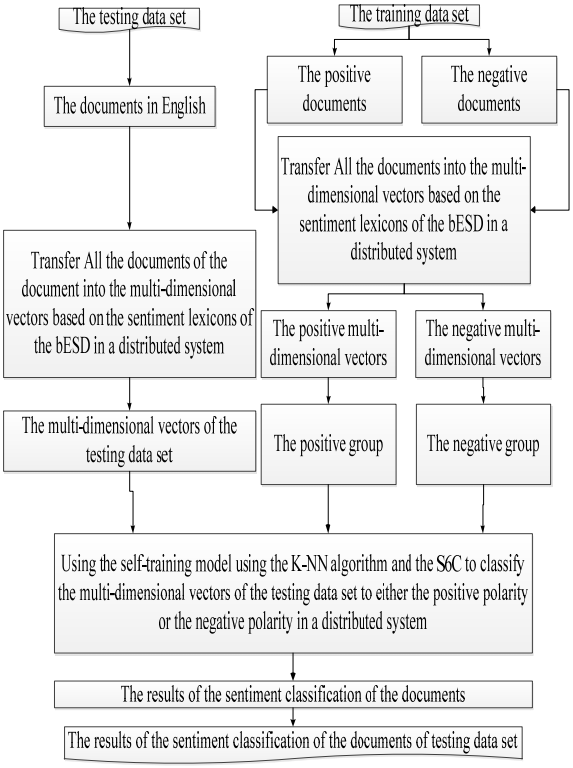

Figure 12: Overview of our novel model.

According to the researches related to a K-Nearest Neighbors algorithm (K-NN) in [45-49], we present the K-NN algorithm which is enhanced to be able to classify the sentiments (positive, negative, or neutral) for the documents. The main ideas of the K-NN are as follows:

1. Identify the K parameter (K - Nearest Neighbors): in this survey, we choose K = 2;
2. Calculate the distance between the vector (which need to be clustered) with the vectors in the training data by Euclidean distance;
3. Arrange the distances by acending order; and Identify K - nearest neighbors with the vectors which need to be clustered;
4. Get all clusters of K - nearest neighbors which are identified;
5. Based on all clusters of nearest neighbors to identify the cluster of the vector;

In Figure 13, we use the self-training model using the K-NN algorithm and the S6C to classify the documents of the testing data set into either the positive or the negative in the distributed system.
In Figure 13, this stage comprises two phases: the Hadoop Map phase and the Hadoop Reduce phase.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The input of the Hadoop Map is the documents of the testing data set and the training data set. The output of the Hadoop Map is the results of the sentiment classification of 100 multi-dimensional vectors. The input of the Hadoop Reduce is the output of the Hadoop Map, thus, the input of the Hadoop Reduce is the results of the sentiment classification of 100 multi-dimensional vectors. The output of the Hadoop Reduce is the results of the sentiment classification of the documents of the testing data set.

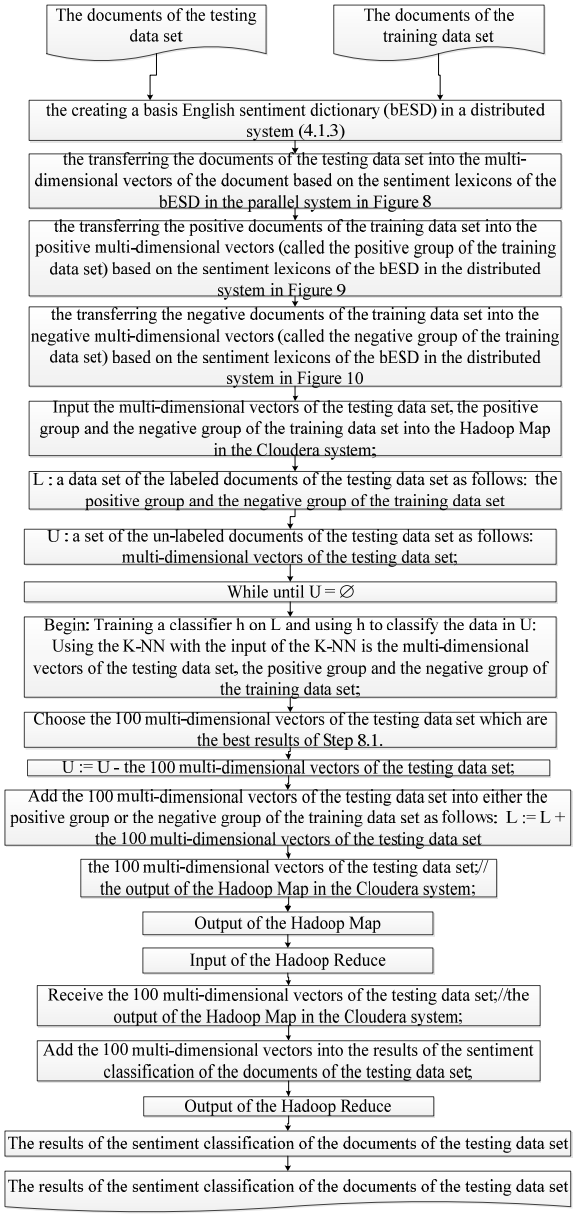

Figure 13: Overview of using the self-training model using the K-NN algorithm and the S6C to classify the documents of the testing data set into either the positive or the negative in the distributed system.

We build the algorithm 19 to performing the Hadoop Map phase of classifying all the documents of the testing data set into either the positive or the negative in the sequential system by using the self-training model with the K-NN algorithm and the S6C. The main ideas of the algorithm 19 are as follows:

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

Output: the results of the sentiment classification of 100 documents of the testing data set (positive, negative, or neutral); //the output of the Hadoop Map in the Cloudera system;

Step 1: the creating a basis English sentiment dictionary (bESD) in a distributed system (4.1.3)

Step 2: the transferring the documents of the testing data set into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system in Figure 8

Step 3: the transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive group of the training data set) based on the sentiment lexicons of the bESD in the distributed system in Figure 9

Step 4: the transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative group of the training data set) based on the sentiment lexicons of the bESD in the distributed system in Figure 10

Step 5: Input the multi-dimensional vectors of the testing data set, the positive group and the negative group of the training data set into the Hadoop Map in the Cloudera system;

Step 6: L : a data set of the labeled documents of the testing data set as follows: the positive group and the negative group of the training data set

Step 7: U : a set of the un-labeled documents of the training data set as follows: the multi-dimensional vectors of the testing data set;

Step 8: While until U = ∅

Step 8.1: Begin: Training a classifier h on L and using h to classify the data in U: Using the K-NN with the input of the K-NN is the multi-dimensional vectors of the testing data set, the positive group and the negative group of the training data set;

Step 8.2: Choose the 100 multi-dimensional vectors of the testing data set which are the best results of Step 8.1.

Step 8.3: U := U - the 100 multi-dimensional vectors of the testing data set;

Step 8.4: Add the 100 multi-dimensional vectors of the testing data set into either the positive group or

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the negative group of the training data set as follows: L := L + the 100 multi-dimensional vectors of the testing data set

Step 8.5: Return the 100 multi-dimensional vectors of the testing data set;//the output of the Hadoop Map in the Cloudera system;

We propose the algorithm 20 to implementing the Hadoop Reduce phase of classifying all the documents of the testing data set into either the positive or the negative in the sequential system by using the self-training model with the K-NN algorithm and the S6C. The main ideas of the algorithm 20 are as follows:

Input: the 100 multi-dimensional vectors of the testing data set;//the output of the Hadoop Map in the Cloudera system;

Output: the results of the sentiment classification of the documents of the testing data set (positive, negative, or neutral);

Step 1: Receive the 100 multi-dimensional vectors of the testing data set;//the output of the Hadoop Map in the Cloudera system;

Step 2: Add the 100 multi-dimensional vectors into the results of the sentiment classification of the documents of the testing data set;

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

5. EXPERIMENT

We have measured an Accuracy (A) to calculate the accuracy of the results of emotion classification.

We used a Java programming language for programming to save data sets, implementing our proposed model to classify the 12,500,000 documents of the testing data set and the 2,000 documents of the training data set. To implement the proposed model, we have already used Java programming language to save the English testing data set and to save the results of emotion classification.

The proposed model was implemented in both the sequential system and the distributed network environment.

Our model related to the self-training model using the K-NN algorithm and the S6C is implemented in the sequential environment with the configuration as follows: The sequential environment in this research includes 1 node (1 server). The configuration of the server in the sequential environment is: Intel® Server Board S1200V3RPS, Intel® Pentium® Processor G3220 (3M Cache, 3.00 GHz), 2GB CC3-10600 ECC 1333 MHz LP Unbuffered DIMMs. The operating system of the server is: Cloudera. The Java language is used in programming our model related to the self-training model using the K-NN algorithm and the S6C.

The novel model related to the self-training model using the K-NN algorithm and the S6C is performed in the Cloudera parallel network environment with the configuration 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 Cache, 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 self-training model using the K-NN algorithm and the S6C in the Cloudera

In Table 5, we show the results of the documents in the testing data set.

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

In Table 7, we display the average time of the classification of our new model for the documents in testing data set.

6. CONCLUSION

In this survey, a new model has been proposed to classify sentiment of many documents in English using the self-training model using the K-NN algorithm and the S6C with Hadoop Map (M) /Reduce (R) in the Cloudera parallel network environment. Based on our proposed new model, we have achieved 89.13% accuracy of the testing data set in Table 6. Until now, not many studies have shown that the clustering methods can be used to classify data. Our research shows that clustering methods are used to classify data and, in particular, can be used to classify the sentiments (positive, negative, or neutral) in text.

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

According to Table 7, the average time of the sentiment classification of using the self-training model using the K-NN algorithm and the S6C in the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

sequential environment is 49,537,581 seconds / 12,500,000 English documents and it is greater than the average time of the sentiment classification of using the self-training model using the K-NN algorithm and the S6C in the Cloudera parallel network environment with 3 nodes which is 15,179,193 seconds / 12,500,000 English documents. The average time of the sentiment classification of using the self-training model using the K-NN algorithm and the S6C in the Cloudera parallel network environment with 9 nodes is 5,426,397 seconds / 12,500,000 English documents, and it is the shortest time in the table. Besides, the average time of the sentiment classification of using the self-training model using the K-NN algorithm and the S6C in the Cloudera parallel network environment with 6 nodes is 6,789,596 seconds / 12,500,000 English documents

The execution time of using the self-training model using the K-NN algorithm and the S6C 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 accuracy of the proposed model is depending on many factors as follows:

1)The self-training – related algorithms
2)The testing data set
3)The documents of the testing data set must be standardized carefully.
4)Transferring one document into one multi-dimensional vector based on the sentiment lexicons.
5)The S6C – related equations.
6)The K-NN – related algorithms.

The execution time of the proposed model is depending on many factors as follows:

1)The parallel network environment such as the Cloudera system.
2)The distributed functions such as Hadoop Map (M) and Hadoop Reduce (R).
3)The self-training – related algorithms
4)The performance of the distributed network system.
5)The number of nodes of the parallel network environment.
6)The performance of each node (each server) of the distributed environment.
7)The sizes of the training data set and the testing data set.
8)Transferring one document into one multi-dimensional vector according to the sentiment lexicons.
9)The Google search engine.
10)The S6C – related equations.
11)The K-NN – related algorithms.

The proposed model has many advantages and disadvantages. Its positives are as follows: It uses The self-training model using the K-NN algorithm and the S6C with the multi-dimensional vectors based on the sentiment lexicons to classify the sentiments of English documents based on sentences. The proposed model can process millions of documents in the shortest time. This study can be performed in distributed systems to shorten the execution time of the proposed model. It can be applied to other languages. Its negatives are as follows: It has a low rate of accuracy. It costs too much and takes too much time to implement this proposed model.

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

In Table 8, we display the comparisons of our model's results with the works related to the K-Nearest Neighbors algorithm (K-NN) in [45-49].

The comparisons of our model's benefits and drawbacks with the studies related to the K-Nearest Neighbors algorithm (K-NN) in [45-49] are shown in Table 9.

In Table 10, we present the comparisons of our model's results with the works related to the Self-Training algorithm in [50-54]

The comparisons of our model's benefits and drawbacks with the studies related to the Self-Training algorithm in [50-54] are displayed in Table 11.

In Table 12, we show the comparisons of our model's results with the works in [55-57]

The comparisons of our model's advantages and disadvantages with the works in [55-57] are presented in Table 13.

In Table 14, we display the comparisons of our model with the latest sentiment classification models (or the latest sentiment classification methods) in [58-70].

The comparisons of our model's positives and negatives the latest sentiment classification models (or the latest sentiment classification methods) in [58-70] are shown in Table 15.

FUTURE WORK

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

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and speeches, and machines that can analyze sentiments.

REFRENCES:

[1] Aleksander Bai, Hugo Hammer, "Constructing sentiment lexicons in Norwegian from a large text corpus", 2014 IEEE 17th International Conference on Computational Science and Engineering, 2014.

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

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

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

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

[6] Amal Htait, Sebastien Fournier, Patrice Bellot, "LSIS at SemEval-2016 Task 7: Using Web Search Engines for English and Arabic Unsupervised Sentiment Intensity Prediction", Proceedings of SemEval-2016, California, 2016, pp 481–485.

[7] Xiaojun Wan, "Co-Training for Cross-Lingual Sentiment Classification", Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, Singapore, 2009, pp 235–243.

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

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

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

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

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

[13]Guangwei Wang, Kenji Araki, "Modifying SO-PMI for Japanese Weblog Opinion Mining by Using a Balancing Factor and Detecting Neutral Expressions", Proceedings of NAACL HLT 2007, Companion Volume, NY, 2007, pp 189–192.

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

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

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

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

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

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

[20]Yong REN, Nobuhiro KAJI, Naoki YOSHINAGA, Masaru KITSUREGAW, "Sentiment Classification in Under-Resourced Languages Using Graph-based Semi-supervised Learning Methods", IEICE TRANS. INF. & SYST., VOL.E97–D, NO.4, DOI: 10.1587/transinf.E97.D.1, 2014.

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

[22]Yong Ren, Nobuhiro Kaji, Naoki Yoshinaga, Masashi Toyoda, Masaru Kitsuregawa, "Sentiment Classification in Resource-Scarce Languages by using Label Propagation", Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation, Institute of Digital Enhancement of Cognitive Processing, Waseda University, 2011, pp 420 - 429.

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

[24]Julia V. Ponomarenko, Philip E. Bourne, Ilya N. Shindyalov, "Building an automated classification of DNA-binding protein domains", BIOINFORMATICS, Vol. 18, 2002, pp S192-S201.

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

[26]Snežana Mladenović Drinić, Ana Nikolić, Vesna Perić, "Cluster Analysis Of Soybean Genotypes Based On RAPD Markers", Proceedings 43rd Croatian And 3rd International Symposium On Agriculture. Opatija. Croatia, 2008, 367- 370.

[27]Tamás, Júlia; Podani, János; Csontos, Péter, "An extension of presence/absence coefficients to abundance data:a new look at absence",Journal of Vegetation Science 12: 401-410, 2001.

[28]Vo Ngoc Phu, Vo Thi Ngoc Chau, Vo Thi Ngoc Tran, Nguyen Duy Dat, "A Vietnamese adjective emotion dictionary based on exploitation of Vietnamese language characteristics", International Journal of Artificial Intelligence Review (AIR), doi:10.1007/s10462-017-9538-6, 2017, 67 pages.

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

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

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

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

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

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

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

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

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

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

[39]Seung-Seok Choi, Sung-Hyuk Cha, Charles C. Tappert, "A Survey Of Binary Similarity And Distance Measures", Systemics, Cybernetics And Informatics, Issn: 1690-4524, Volume 8 - Number 1, 2010

[40]Kyoung Nam Kim, Il Hwan Shin, Jae Yun Sung, Baek Soo Kwak, Hyung Bin Lim, Young Joon Jo, Jung Yeul Kim, "The effect of center point shift on the measurement of macular thickness: a spectral domain-optical coherence tomography study", Graefe's Archive for

Clinical and Experimental Ophthalmology, Volume 255, Issue 6, 2017, pp 1107–1113

[41]Rodham E. Tulloss, "Assessment of Similarity Indices for Undesirable Properties and a new Tripartite Similarity Index Based on Cost Functions", Offprint from Palm, M. E. and I. H. Chapela, eds. 1997. MS6Cology in Sustainable Development: Expanding Concepts, Vanishing Borders. (Parkway Publishers, Boone, North Carolina): 122-143, 1997

[42]Allison H. Squiresa, W. E. Moernera, "Direct single-molecule measurements of phycocyanobilin photophysics in monomeric C-phycocyanin", Allison H. Squires, 9779–9784, doi: 10.1073/pnas.1705435114, 2017

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

[44]Benyuan Ma, Bin Wang, Wei Zhang, Nian Wei, Tiecheng Lu, Junbao He, "Promotion of powder crystallinity and its influence on the properties of Nd:YAG transparent ceramics", Optical Materials, Volume 64, Pages 384-390, https://doi.org/10.1016/j.optmat.2017.01.006,2017

[45]J. M. Keller; M. R. Gray; J. A. Givens, "A fuzzy K-nearest neighbor algorithm", IEEE Transactions on Systems, Man, and Cybernetics, Volume: SMC-15, Issue: 4, DOI: 10.1109/TSMC.1985.6313426,1985

[46]Ludmila I. Kuncheva, "Editing for the k-nearest neighbors rule by a genetic algorithm", Pattern Recognition Letters, Volume 16, Issue 8, Pages 809-814, https://doi.org/10.1016/0167-8655(95)00047-K,1995

[47]X. Yu; K.Q. Pu; N. Koudas, "Monitoring k-nearest neighbor queries over moving objects", Proceedings. 21st International Conference on Data Engineering, 2005. ICDE 2005, DOI: 10.1109/ICDE.2005.92, Tokoyo, Japan, Japan, 2005

[48]Eui-Hong (Sam) Han, George Karypis, Vipin Kumar, "Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification", Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD 2001: Advances in Knowledge Discovery and Data Mining, 2011, pp 53-65

[49]K. Mouratidis; D. Papadias; S. Bakiras; Yufei Tao, "A threshold-based algorithm for continuous monitoring of k nearest neighbors", IEEE Transactions on Knowledge and Data Engineering, Volume: 17, Issue: 11, DOI: 10.1109/TKDE.2005.172,2005

[50]Chuck Rosenberg, Martial Hebert, Henry Schneiderman, "Semi-Supervised Self-Training of Object Detection Models", Seventh IEEE Workshop on Applications of Computer Vision, 2005

51. Yuanqing Li, Cuntai Guan, Huiqi Li, Zhengyang Chin (2008) A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system. Pattern Recognition Letters, Volume 29, Issue 9, Pages 1285-1294, https://doi.org/10.1016/j.patrec.2008.01.030

[52]Alexandre Lomsadze, Vardges Ter-Hovhannisyan, Yury O. Chernoff, Mark Borodovsky, "Gene identification in novel eukaryotic genomes by self-training algorithm", Nucleic Acids Research, Volume 33, Issue 20, Pages 6494–6506, https://doi.org/10.1093/nar/gki937, 2005

[53]John Besemer, Alexandre Lomsadze, Mark Borodovsky, "GeneMarkS: a self-training method for prediction of gene starts in microbial genomes", Implications for finding sequence motifs in regulatory regions. Nucleic Acids Research, Volume 29, Issue 12, 2001, Pages 2607–2618, https://doi.org/10.1093/nar/29.12.2607

[54]Wei Hu Nanjing, Jianfeng Chen, Yuzhong Qu, "A self-training approach for resolving object coreference on the semantic web", WWW '11 Proceedings of the 20th international conference on World wide web, Pages 87-96, Hyderabad, India, 2011

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

[56]Víctor Carrera-Trejo, Grigori Sidorov, Sabino Miranda-Jiménez, Marco Moreno Ibarra and Rodrigo Cadena Martínez, "Latent Dirichlet Allocation complement in the vector space model for Multi-Label Text Classification", International Journal of Combinatorial Optimization Problems and Informatics, Vol. 6, No. 1, 2015, pp. 7-19.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[57]Pascal Soucy, Guy W. Mineau, "Beyond TFIDF Weighting for Text Categorization in the Vector Space Model", Proceedings of the 19th International Joint Conference on Artificial Intelligence, pp. 1130-1135, USA, 2015

[58]Basant Agarwal, Namita Mittal, "Machine Learning Approach for Sentiment Analysis", Prominent Feature Extraction for Sentiment Analysis, Print ISBN 978-3-319-25341-1, 10.1007/978-3-319-25343-5_3, 21-45, 2016

[59]Basant Agarwal, Namita Mittal, "Semantic Orientation-Based Approach for Sentiment Analysis", Prominent Feature Extraction for Sentiment Analysis, Print ISBN 978-3-319-25341-1, 10.1007/978-3-319-25343-5_6, 77-88, 2016

[60]Sérgio Canuto, Marcos André, Gonçalves, Fabrício Benevenuto, "Exploiting New Sentiment-Based Meta-level Features for Effective Sentiment Analysis", Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM '16), 53-62, New York USA, 2016

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

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

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

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

[65]Vo Ngoc Phu, Nguyen Duy Dat, Vo Thi Ngoc Tran, Vo Thi Ngoc Tran, "Fuzzy C-Means for English Sentiment Classification in a Distributed System", International Journal of Applied Intelligence (APIN), DOI: 10.1007/s10489-016-0858-z, 1-22, November 2016.

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

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

[68]Vo Ngoc Phu, Vo Thi Ngoc Tran, Vo Thi Ngoc Chau, Nguyen Duy Dat, Khanh Ly Doan Duy, "A Decision Tree using ID3 Algorithm for English Semantic Analysis", International Journal of Speech Technology (IJST), DOI: 10.1007/s10772-017-9429-x, 2017, 23 pages

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

APPENDICES

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

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

Table 3: Comparisons of our model's results with the works related to the S6 coefficient (S6C) in [39, 44].

Table 4: Comparisons of our model's benefits and drawbacks with the studies related to the S6 coefficient (S6C) in [39, 44].

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

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

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

Table 8: Comparisons of our model's results with the works related to the K-Nearest Neighbors algorithm (K-NN) in [45-49]

Table 9: Comparisons of our model's benefits and drawbacks with the studies related to the K-Nearest Neighbors algorithm (K-NN) in [45-49]

Table 10: Comparisons of our model's results with the works related to the Self-Training algorithm in [50-54]

Table 11: Comparisons of our model's benefits and drawbacks with the studies related to the Self-Training algorithm in [50-54]

Table 12: Comparisons of our model's results with the works in [55-57]

Table 13: Comparisons of our model's advantages and disadvantages with the works in [55-57]

Table 14: Comparisons of our model with the latest sentiment classification models (or the latest sentiment classification methods) in [58-70]

Table 15: Comparisons of our model's positives and negatives the latest sentiment classification models (or the latest sentiment classification methods) in [58-70]

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

S6 coefficient (S6C)

Semantic classification, sentiment classification: SC

Clustering technique: CT.

Parallel network system: PNS (distributed system).

Special Domain: SD.

Depending on the training data set: DT.

Vector Space Model: VSM

No Mention: NM

English Language: EL.

Studies	PMI	JM	Language	SD	DT	S6C	SC	Other measures	Search engines
[1]	Yes	No	English	Yes	Yes	No	Yes	No	No Mention
[2]	Yes	No	English	Yes	No	No	Yes	Latent Semantic Analysis (LSA)	AltaVista
[3]	Yes	No	English	Yes	Yes	No	Yes	Baseline; Turney-inspired; NB; Cluster+NB; Human	AltaVista
[4]	Yes	No	English German	Yes	Yes	No	Yes	SimRank	Google search engine
[5]	Yes	No	English Macedonian	Yes	Yes	No	Yes	No Mention	AltaVista search engine
[6]	Yes	No	English Arabic	Yes	No	No	Yes	No Mention	Google search engine Bing search engine
[7]	Yes	No	English	Yes	Yes	No	Yes	SVM(CN); SVM(EN); SVM(ENCN1);	No Mention

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

			Chinese					SVM(ENCN2); TSVM(CN); TSVM(EN); TSVM(ENCN1); TSVM(ENCN2); CoTrain	
[8]	Yes	No	English Spanish	Yes	Yes	No	Yes	SO Calculation SVM	Google
[9]	Yes	No	Chinese Tibetan	Yes	Yes	No	Yes	- Feature selection -Expectation Cross Entropy -Information Gain	No Mention
[10]	Yes	No	Chinese	Yes	Yes	No	Yes	DF, CHI, MI andIG	No Mention
[11]	Yes	No	Chinese	Yes	No	No	Yes	Information Bottleneck Method (IB); LE	AltaVista
[12]	Yes	No	Chinese	Yes	Yes	No	Yes	SVM	Google Yahoo Baidu
[13]	Yes	No	Japanese	No	No	No	Yes	Harmonic−Mean	Google and replaced the NEAR operator with the AND operator inthe SO formula.
[14]	Yes	Yes	English	Yes	Yes	No	Yes	Dice; NGD	Google search engine
[15]	Yes	Yes	English	Yes	No	No	Yes	Dice; Overlap	Google
[16]	No	Yes	English	Yes	Yes	No	Yes	A Jaccard index based clustering algorithm (JIBCA)	No Mention
[17]	No	Yes	English	Yes	Yes	No	Yes	Naıve Bayes, Two-Step Multinomial Naıve Bayes, and Two-Step Polynomial-Kernel Support Vector Machine	Google
[18]	No	Yes	Arabic	No	No	No	Yes	Naive Bayes (NB); Support Vector Machines (SVM); RoS6Chio; Cosine	No Mention
[19]	No	Yes	Chinese	Yes	Yes	No	Yes	A new score–Economic Value (EV), etc.	Chinese search
[20]	No	Yes	Chinese	Yes	Yes	No	Yes	Cosine	No Mention
[21]	No	Yes	English	No	Yes	No	Yes	Cosine	No Mention
[22]	No	Yes	Chinese	No	Yes	No	Yes	Dice; overlap; Cosine	No Mention
[28]	No	No	Vietname se	No	No	No	Yes	Ochiai Measure	Google
[29]	No	No	English	No	No	No	Yes	Cosine coefficient	Google
[30]	No	No	English	No	No	No	Yes	Sorensen measure	Google
[31]	No	Yes	Vietname se	No	No	No	Yes	Jaccard	Google
[32]	No	No	English	No	No	No	Yes	Tanimoto coefficient	Google
Our work	No	No	English Language	Yes	Yes	Yes	Yes	No	Google search engine

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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

Surveys	Approach	Advantages	Disadvantages
[1]	Constructing sentiment lexicons in Norwegian from a large text corpus	Through the authors' PMI computations in this survey they used a distance of 100 words from the seed word, but it might be that other lengths that generate better sentiment lexicons. Some of the authors' preliminary research showed that 100 gave a better result.	The authors need to investigate this more closely to find the optimal distance. Another factor that has not been investigated much in the literature is the selection of seed words. Since they are the basis for PMI calculation, it might be a lot to gain by finding better seed words. The authors would like to explore the impact that different approaches to seed word selection have on the performance of the developed sentiment lexicons.
[2]	Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus.	This survey has presented a general strategy for learning semantic orientation from semantic association, SO-A. Two instances of this strategy have been empirically evaluated, SO-PMI-IR and SO-LSA. The accuracy of SO-PMI-IR is comparable to the accuracy of HM, the algorithm of Hatzivassiloglou and McKeown (1997). SO-PMI-IR requires a large corpus, but it is simple, easy to implement, unsupervised, and it is not restricted to adjectives.	No Mention
[3]	Graph-based user classification for informal online political discourse	The authors describe several experiments in identifying the political orientation of posters in an informal environment. The authors' results indicate that the most promising approach is to augment text classification methods by exploiting information about how posters interact with each other	There is still much left to investigate in terms of optimizing the linguistic analysis, beginning with spelling correction and working up to shallow parsing and co-reference identification. Likewise, it will also be worthwhile to further investigate exploiting sentiment values of phrases and clauses, taking cues from methods
[4]	A novel, graph-based approach using SimRank.	The authors presented a novel approach to the translation of sentiment information that outperforms SOPMI, an established method. In particular, the authors could show that SimRank outperforms SO-PMI for values of the threshold x in an interval that most likely leads to the correct separation of positive, neutral, and negative adjectives.	The authors' future work will include a further examination of the merits of its application for knowledge-sparse languages.
[5]	Analysis in Twitter for Macedonian	The authors' experimental results show an F1-score of 92.16, which is very strong and is on par with the best results for English, which were achieved in recent SemEval competitions.	In future work, the authors are interested in studying the impact of the raw corpus size, e.g., the authors could only collect half a million tweets for creating lexicons and analyzing/evaluating the system, while Kiritchenko et al. (2014) built their lexicon on million tweets and evaluated their system on 135 million English tweets. Moreover, the authors are interested not only in quantity but also in quality, i.e., in studying the quality of the individual words and phrases used as seeds.
[6]	Using Web Search Engines for English and Arabic Unsupervised Sentiment Intensity	- For the General English sub-task, the authors' system has modest but interesting results. - For the Mixed Polarity English sub-task, the authors' system results achieve the second place. - For the Arabic phrases sub-task, the authors' system has very interesting results since they applied the unsupervised method only	Although the results are encouraging, further investigation is required, in both languages, concerning the choice of positive and negative words which once associated to a phrase, they make it more negative or more positive.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	Prediction		
[7]	Co-Training for Cross-Lingual Sentiment Classification	The authors propose a co-training approach to making use of unlabeled Chinese data. Experimental results show the effectiveness of the proposed approach, which can outperform the standard inductive classifiers and the transductive classifiers.	In future work, the authors will improve the sentiment classification accuracy in the following two ways: 1) The smoothed co-training approach used in (Mihalcea, 2004) will be adopted for sentiment classification. 2) The authors will employ the structural correspondence learning (SCL) domain adaption algorithm used in (Blitzer et al., 2007) for linking the translated text and the natural text.
[8]	Cross-Linguistic Sentiment Analysis: From English to Spanish	Our Spanish SO calculator (SOCAL) is clearly inferior to the authors' English SO-CAL, probably the result of a number of factors, including a small, preliminary dictionary, and a need for additional adaptation to a new language. Translating our English dictionary also seems to result in significant semantic loss, at least for original Spanish texts.	No Mention
[9]	Micro–blog Emotion Orientation Analysis Algorithm Based on Tibetan and Chinese Mixed Text	By emotion orientation analyzing and studying of Tibetan microblog which is concerned in Sina, making Tibetan Chinese emotion dictionary, Chinese sentences, Tibetan part of speech sequence and emotion symbol as emotion factors and using expected cross entropy combined fuzzy set to do feature selection to realize a kind of microblog emotion orientation analyzing algorithm based on Tibetan and Chinese mixed text. The experimental results showed that the method can obtain better performance in Tibetan and Chinese mixed Microblog orientation analysis.	No Mention
[10]	An empirical study of sentiment analysis for Chinese documents	Four feature selection methods (MI, IG, CHI and DF) and five learning methods (centroid classifier, K-nearest neighbor, winnow classifier, Naïve Bayes and SVM) are investigated on a Chinese sentiment corpus with a size of 1021 documents. The experimental results indicate that IG performs the best for sentimental terms selection and SVM exhibits the best performance for sentiment classification. Furthermore, the authors found that sentiment classifiers are severely dependent on domains or topics.	No Mention
[11]	Adapting Information Bottleneck Method for Automatic Construction of Domain-oriented Sentiment Lexicon	The authors' theory verifies the convergence property of the proposed method. The empirical results also support the authors' theoretical analysis. In their experiment, it is shown that proposed method greatly outperforms the baseline methods in the task of building out-of-domain sentiment lexicon.	In this study, only the mutual information measure is employed to measure the three kinds of relationship. In order to show the robustness of the framework, the authors' future effort is to investigate how to integrate more measures into this framework.
[12]	Sentiment Classification for Consumer Word-of-Mouth in Chinese: Comparison between Supervised and Unsupervised Approaches	This study adopts three supervised learning approaches and a web-based semantic orientation approach, PMI-IR, to Chinese reviews. The results show that SVM outperforms naive bayes and N-gram model on various sizes of training examples, but does not obviously exceeds the semantic orientation approach when the number of training examples is smaller than 300.	No Mention
[13]	Modifying SO-PMI for Japanese Weblog Opinion Mining by Using a	After these modifications, the authors achieved a well-balanced result: both positive and negative accuracy exceeded 70%. This shows that the authors' proposed approach not only adapted the	In the future, the authors will evaluate different choices of words for the sets of positive and negative reference words. The authors also plan to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	Balancing Factor and Detecting Neutral Expressions	SO-PMI for Japanese, but also modified it to analyze Japanese opinions more effectively.	appraise their proposal on other languages.
[14]	In this survey, the authors empirically evaluate the performance of different corpora in sentiment similarity measurement, which is the fundamental task for word polarity classification.	Experiment results show that the Twitter data can achieve a much better performance than the Google, Web1T and Wikipedia based methods.	No Mention
[15]	Adjective-Based Estimation of Short Sentence's Impression	The adjectives are ranked and top na adjectives are considered as an output of system. For example, the experiments were carried out and got fairly good results. With the input "it is snowy", the results are white (0.70), light (0.49), cold (0.43), solid (0.38), and scenic (0.37)	In the authors' future work, they will improve more in the tasks of keyword extraction and semantic similarity methods to make the proposed system working well with complex inputs.
[16]	Jaccard Index based Clustering Algorithm for Mining Online Review	In this work, the problem of predicting sales performance using sentiment information mined from reviews is studied and a novel JIBCA Algorithm is proposed and mathematically modeled. The outcome of this generates knowledge from mined data that can be useful for forecasting sales.	For future work, by using this framework, it can extend it to predicting sales performance in the other domains like customer electronics, mobile phones, computers based on the user reviews posted on the websites, etc.
[17]	Twitter sentiment classification for measuring public health concerns	Based on the number of tweets classified as Personal Negative, the authors compute a Measure of Concern (MOC) and a timeline of the MOC. We attempt to correlate peaks of the MOC timeline to the peaks of the News (Non-Personal) timeline. The authors' best accuracy results are achieved using the two-step method with a Naïve Bayes classifier for the Epidemic domain (six datasets) and the Mental Health domain (three datasets).	No Mention
[18]	Ensemble of Classification algorithms for Subjectivity and Sentiment Analysis of Arabic Customers' Reviews	The experimental results show that the ensemble of the classifiers improves the classification effectiveness in terms of macro-F1 for both levels. The best results obtained from the subjectivity analysis and the sentiment classification in terms of macro-F1 are 97.13% and 90.95% respectively.	No Mention
[19]	Automatic Construction of Financial Semantic Orientation Lexicon from Large-Scale Chinese News Corpus	Semantic orientation lexicon of positive and negative words is indispensable for sentiment analysis. However, many lexicons are manually created by a small number of human subjects, which are susceptible to high cost and bias. In this survey, the authors propose a novel idea to construct a financial semantic orientation lexicon from large-scale Chinese news corpus automatically ...	No Mention
[20]	Sentiment Classification in Under-Resourced Languages Using Graph-based Semi-supervised Learning Methods	In particular, the authors found that choosing initially labeled vertices in aS6Cordance with their degree and PageRank score can improve the performance. However, pruning unreliable edges will make things more difficult to predict. The authors believe that other people who are interested in this field can benefit from their empirical findings.	As future work, first, the authors will attempt to use a sophisticated approach to induce better sentiment features. The authors consider such elaborated features improve the classification performance, especially in the book domain. The authors also plan to exploit a much larger amount

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

			of unlabeled data to fully take advantage of SSL algorithms
[21]	A text-mining approach and combine it with semantic network analysis tools	In summary, the authors hope the text-mining and derived market-structure analysis presented in this paper provides a first step in exploring the extremely large, rich, and useful body of consumer data readily available on Web 2.0.	No Mention
[22]	Sentiment Classification in Resource-Scarce Languages by using Label Propagation	The authors compared our method with supervised learning and semi-supervised learning methods on real Chinese reviews classification in three domains. Experimental results demonstrated that label propagation showed a competitive performance against SVM or Transductive SVM with best hyper-parameter settings. Considering the difficulty of tuning hyper-parameters in a resourcescarce setting, the stable performance of parameter-free label propagation is promising.	The authors plan to further improve the performance of LP in sentiment classification, especially when the authors only have a small number of labeled seeds. The authors will exploit the idea of restricting the label propagating steps when the available labeled data is quite small.
[28]	A Vietnamese adjective emotion dictionary based on exploitation of Vietnamese language characteristics	The Vietnamese adjectives often bear emotion which values (or semantic scores) are not fixed and are changed when they appear in different contexts of these phrases. Therefore, if the Vietnamese adjectives bring sentiment and their semantic values (or their sentiment scores) are not changed in any context, then the results of the emotion classification are not high accuracy. The authors propose many rules based on Vietnamese language characteristics to determine the emotional values of the Vietnamese adjective phrases bearing sentiment in specific contexts. The authors' Vietnamese sentiment adjective dictionary is widely used in applications and researches of the Vietnamese semantic classification.	not calculating all Vietnamese words completely; not identifying all Vietnamese adjective phrases fully, etc.
[29]	A Valences-Totaling Model for English Sentiment Classification	The authors present a full range of English sentences; thus, the emotion expressed in the English text is classified with more precision. The authors new model is not dependent on a special domain and training data set—it is a domain-independent classifier. The authors test our new model on the Internet data in English. The calculated valence (and polarity) of English semantic words in this model is based on many documents on millions of English Web sites and English social networks.	It has low accuracy; it misses many sentiment-bearing English words; it misses many sentiment-bearing English phrases because sometimes the valence of a English phrase is not the total of the valences of the English words in this phrase; it misses many English sentences which are not processed fully; and it misses many English documents which are not processed fully.
[30]	Shifting Semantic Values of English Phrases for Classification	The results of the sentiment classification are not high accuracy if the English phrases bring the emotions and their semantic values (or their sentiment scores) are not changed in any context. For those reasons, the authors propose many rules based on English language grammars to calculate the sentimental values of the English phrases bearing emotion in their specific contexts. The results of this work are widely used in applications and researches of the English semantic classification.	This survey is only applied to the English adverb phrases. The proposed model is needed to research more and more for the different types of the English words such as English noun, English adverbs, etc
[31]	A Valence-Totaling Model for Vietnamese Sentiment Classification	The authors have used the VTMfV to classify 30,000 Vietnamese documents which include the 15,000 positive Vietnamese documents and the 15,000 negative Vietnamese documents. The authors have achieved accuracy in 63.9% of the authors' Vietnamese testing data set. VTMfV is not dependent on the special domain. VTMfV is also	it has a low accuracy.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

		not dependent on the training data set and there is no training stage in this VTMfV. From the authors' results in this work, our VTMfV can be applied in the different fields of the Vietnamese natural language processing. In addition, the authors' TCMfV can be applied to many other languages such as Spanish, Korean, etc. It can also be applied to the big data set sentiment classification in Vietnamese and can classify millions of the Vietnamese documents	
[32]	Semantic Lexicons of English Nouns for Classification	The proposed rules based on English language grammars to calculate the sentimental values of the English phrases bearing emotion in their specific contexts. The results of the sentiment classification are not high accuracy if the English phrases bring the emotions and their semantic values (or their sentiment scores) are not changed in any context. The valences of the English words (or the English phrases) are identified by using Tanimoto Coefficient (TC) through the Google search engine with AND operator and OR operator. The emotional values of the English noun phrases are based on the English grammars (English language characteristics)	This survey is only applied in the English noun phrases. The proposed model is needed to research more and more about the different types of the English words such as English English adverbs, etc.
Our work	-We use the Self-training model based on the K-NN algorithm and the multi-dimensional vectors to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. -The sentiment lexicons of the bESD are based on the S6 coefficient (S6C) through the Google search engine with AND operator and OR operator. -The novel model uses the multi-dimensional vectors according to the sentiment lexicons. The advantages and disadvantages of this survey are shown in the Conclusion section.		

Table 3: Comparisons of our model's results with the works related to the S6 coefficient (S6C) in [39-44].

Studies	PMI	JM	S6 coefficient (S6C)	Language	SD	DT	Sentiment Classification
[39]	Yes	Yes	Yes	English	NM	NM	No mention
[40]	No	No	Yes	NM	NM	NM	No mention
[41]	No	No	Yes	NM	NM	NM	No mention
[42]	No	No	Yes	NM	NM	NM	No mention
[43]	No	No	Yes	NM	NM	NM	No mention
[44]	No	No	Yes	NM	NM	NM	No mention
Our work	No	No	Yes	English Language	Yes	Yes	Yes

Table 4: Comparisons of our model's benefits and drawbacks with the studies related to the S6 coefficient (S6C) in [39-44].

Surveys	Approach	Benefits	Drawbacks
[39]	A Survey of Binary Similarity and Distance Measures	Applying appropriate measures results in more aS6Curate data analysis. Notwithstanding, few comprehensive surveys on binary measures have been conducted. Hence the authors collected 76 binary similarity and distance measures used over the last century and reveal their correlations through the hierarchical clustering technique	No mention

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[40]	The effect of center point shift on the measurement of macular thickness: a spectral domain-optical coherence tomography study	When the displacement distance between the measurement center point and the foveal center was within 117.4 μm horizontally and 141.6 μm vertically, the macular thickness measurements did not show any significant differences. However, if the offset of the EDTRS grid center from the anatomic fovea exceeded, the authors noted that the thickness at the fovea increased and the opposite-direction region at the inner circle was significantly thinner than the displaced point.	No mention
[41]	Assessment of Similarity Indices for Undesirable Properties and a new Tripartite Similarity Index Based on Cost Functions	The purpose of this study is to motivate, describe, and offer an implementation for, a working similarity index that avoids the difficulties noted for the others.	No mention
[42]	Direct single-molecule measurements of phycocyanobilin photophysics in monomeric C-phycocyanin	The authors present single-molecule characterization of C-phycocyanin (C-PC), a three-pigment biliprotein that self-assembles to form the midantenna rods of cyanobacterial phycobilisomes. Using the Anti-Brownian Electrokinetic (ABEL) trap to counteract Brownian motion of single particles in real time, the authors directly monitor the changing photophysical states of individual C-PC monomers from Spirulina platensis in free solution by simultaneous readout of their brightness, fluorescence anisotropy, fluorescence lifetime, and emission spectra. These include single-chromophore emission states for each of the three covalently bound phycocyanobilins, providing direct measurements of the spectra and photophysics of these chemically identical molecules in their native protein environment. The authors further show that a simple Förster resonant energy transfer (FRET) network model accurately predicts the observed photophysical states of C-PC and suggests highly variable quenching behavior of one of the chromophores, which should inform future studies of higher-order complexes	No mention
[43]	Finding an appropriate equation to measure similarity between binary vectors: case studies on Indonesian and Japanese herbal medicines	The selection of binary similarity and dissimilarity measures for multivariate analysis is data dependent. The proposed method can be used to find the most suitable binary similarity and dissimilarity equation wisely for a particular data. Our finding suggests that all four types of matching quantities in the Operational Taxonomic Unit (OTU) table are important to calculate the similarity and dissimilarity coefficients between herbal medicine formulas. Also, the binary similarity and dissimilarity measures that include the negative match quantity *d* achieve better capability to separate herbal medicine pairs compared to equations that exclude *d*.	No mention
[44]	Promotion of powder crystallinity and its influence on the properties of Nd:YAG transparent ceramics	The results show that poorly crystallized powder involves a thick layer on particle surface, which is not well crystallized but can escape XRD detection. This poorly crystallized powder contributes more the inhomogeneous ceramic sintering and introduces more the defects in final ceramics, e.g., impure phase inclusions and dislocations. The results reveal that powder crystallinity should be promoted and considered as a further way to improve ceramic properties.	No mention

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Our work	-We use the Self-training model based on the K-NN algorithm and the multi-dimensional vectors to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. -The sentiment lexicons of the bESD are based on the S6 coefficient (S6C) through the Google search engine with AND operator and OR operator. -The novel model uses the multi-dimensional vectors according to the sentiment lexicons. The advantages and disadvantages of this survey are shown in the Conclusion section.

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

	Testing Dataset	Correct Classification	Incorrect Classification
Negative	6,250,000	5,607,124	669,876
Positive	6,250,000	5,534,126	715,874
Summary	12,500,000	11,141,250	1,358,750

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

Proposed Model	Class	Accuracy
Our new model	Negative	89.13%
	Positive	

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

	Average time of the classification / 12,500,000 English documents.
The self-training model using the K-NN algorithm and the S6C in the sequential environment	49,537,581 seconds
The self-training model using the K-NN algorithm and the S6C in the Cloudera distributed system with 3 nodes	15,179,193 seconds
The self-training model using the K-NN algorithm and the S6C in the Cloudera distributed system with 6 nodes	6,789,596 seconds
The self-training model using the K-NN algorithm and the S6C in the Cloudera distributed system with 9 nodes	5,426,397 seconds

Table 8: Comparisons of our model's results with the works related to the K-Nearest Neighbors algorithm (K-NN) in [45-49]

Studies	PMI	JM	K-NN	Language	SD	DT	Sentiment Classification
[45]	Yes	Yes	Yes	English	Yes	Yes	No mention
[46]	No	No	Yes	NM	Yes	Yes	No mention
[47]	No	No	Yes	NM	Yes	Yes	No mention

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[48]	No	No	Yes	NM	Yes	Yes	No mention
[49]	No	No	Yes	NM	Yes	Yes	No mention
Our work	No	No	Yes	English Language	Yes	Yes	Yes

Table 9: Comparisons of our model's benefits and drawbacks with the studies related to the K-Nearest Neighbors algorithm (K-NN) in [45-49]

Surveys	Approach	Benefits	Drawbacks
[45]	A fuzzy K-nearest neighbor algorithm	The K-nearest neighbor decision rule has often been used in these pattern recognition problems. One of the difficulties that arises when utilizing this technique is that each of the labeled samples is given equal importance in deciding the class memberships of the pattern to be classified, regardless of their `typicalness'. The theory of fuzzy sets is introduced into the K-nearest neighbor technique to develop a fuzzy version of the algorithm. Three methods of assigning fuzzy memberships to the labeled samples are proposed, and experimental results and comparisons to the crisp version are presented	No mention
[46]	Editing for the k-nearest neighbors rule by a genetic algorithm	The results are commented together with those obtained with the standard *k*-NN, random selection, Wilson's technique, and the MULTIEDIT algorithm.	No mention
[47]	Monitoring k-nearest neighbor queries over moving objects	The authors relax this assumption, and propose two efficient and scalable algorithms using grid indices. One is based on indexing objects, and the other on queries. For each approach, a cost model is developed, and a detailed analysis along with the respective applicability is presented. The object-indexing approach is further extended to multi-levels to handle skewed data. The authors show by experiments that the authors' grid-based algorithms significantly outperform R-tree-based solutions. Extensive experiments are also carried out to study the properties and evaluate the performance of the proposed approaches under a variety of settings.	No mention
[48]	Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification	The authors present a Weight Adjusted k-Nearest Neighbor (WAKNN) classification that learns feature weights based on a greedy hill climbing technique. The authors also present two performance optimizations of WAKNN that improve the computational performance by a few orders of magnitude, but do not compromise on the classification quality. The authors experimentally evaluated WAKNN on 52 document data sets from a variety of domains and compared its performance against several classification algorithms, such as C4.5, RIPPER, Naive-Bayesian, PEBLS and VSM. Experimental results on these data sets confirm that WAKNN consistently outperforms other existing classification algorithms.	No mention
[49]	A threshold-based algorithm for continuous monitoring of k nearest neighbors	The authros present a threshold-based algorithm for the continuous monitoring of nearest neighbors that minimizes the communication overhead between the server and the data objects. The proposed method can be used with multiple, static, or moving queries, for any distance definition, and does not require additional knowledge (e.g., velocity vectors) besides object locations	No mention
Our work	-We use the Self-training model based on the K-NN algorithm and the multi-dimensional vectors to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. -The sentiment lexicons of the bESD are based on the S6 coefficient (S6C) through the Google search engine with AND operator and OR operator. -The novel model uses the multi-dimensional vectors according to the sentiment lexicons. The advantages and disadvantages of this survey are shown in the Conclusion section.		

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 10: Comparisons of our model's results with the works related to the Self-Training algorithm in [50-54]

Studies	PMI	JM	Self-Training algorithm	Language	SD	DT	Sentiment Classification
[50]	Yes	Yes	Yes	English	Yes	Yes	No mention
[51]	No	No	Yes	NM	Yes	Yes	No mention
[52]	No	No	Yes	NM	Yes	Yes	No mention
[53]	No	No	Yes	NM	Yes	Yes	No mention
[54]	No	No	Yes	NM	Yes	Yes	No mention
Our work	No	No	Yes	English Language	Yes	Yes	Yes

Table 11: Comparisons of our model's benefits and drawbacks with the studies related to the Self-Training algorithm in [50-54]

Surveys	Approach	Benefits	Drawbacks
[50]	Semi-Supervised Self-Training of Object Detection Models	In this work the authors present a semi-supervised approach to training object detection systems based on self-training. The authors implement the authors' approach as a wrapper around the training process of an existing object detector and present empirical results. The key contributions of this empirical study is to demonstrate that a model trained in this manner can achieve results comparable to a model trained in the traditional manner using a much larger set of fully labeled data, and that a training data selection metric that is defined independently of the detector greatly outperforms a selection metric based on the detection confidence generated by the detector.	No mention
[51]	A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system	The authors apply the authors' algorithm to a data set collected from a P300-based brain computer interface (BCI) speller. This algorithm is shown to be able to significantly reduce training effort of the P300-based BCI speller.	No mention
[52]	Gene identification in novel eukaryotic genomes by self-training algorithm	Tests on well-studied eukaryotic genomes have shown that the new method performs comparably or better than conventional methods where the supervised model training precedes the gene prediction step. Several novel genomes have been analyzed and biologically interesting findings are discussed. Thus, a self-training algorithm that had been assumed feasible only for prokaryotic genomes has now been developed for ab initio eukaryotic gene identification.	No mention
[53]	A self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions	The authors have observed that GeneMarkS detects prokaryotic genes, in terms of identifying open reading frames containing real genes, with an accuracy matching the level of the best currently used gene detection methods. Accurate translation start prediction, in addition to the refinement of protein sequence N-terminal data, provides the benefit of precise positioning of the sequence region situated upstream to a gene start. Therefore, sequence motifs related to transcription and translation regulatory sites can be revealed and analyzed with higher precision. These motifs were shown to possess a significant variability, the functional and evolutionary connections of which are discussed.	No mention
[54]	A self-training approach for resolving object coreference on the semantic web	The authors propose a self-training approach for object coreference resolution on the Semantic Web, which leverages the two classes of approaches to bridge the gap between semantically coreferent URIs and potential candidates. For an object URI, the authors firstly establish a kernel that consists of semantically coreferent URIs based on owl:sameAs, (inverse) functional	No mention

		properties and (max-)cardinalities, and then extend such kernel iteratively in terms of discriminative property-value pairs in the descriptions of URIs. In particular, the discriminability is learnt with a statistical measurement, which not only exploits key characteristics for representing an object, but also takes into account the matchability between properties from pragmatics. In addition, frequent property combinations are mined to improve the accuracy of the resolution. The authors implement a scalable system and demonstrate that the authors' approach achieves good precision and recall for resolving object coreference, on both benchmark and large-scale datasets.	
Our work	-We use the Self-training model based on the K-NN algorithm and the multi-dimensional vectors to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. -The sentiment lexicons of the bESD are based on the S6 coefficient (S6C) through the Google search engine with AND operator and OR operator. -The novel model uses the multi-dimensional vectors according to the sentiment lexicons. The advantages and disadvantages of this survey are shown in the Conclusion section.		

Table 12: Comparisons of our model's results with the works in [55-57]

Studies	S6C	CT	Sentiment Classification	PNS	SD	DT	Language	VSM
[55]	No	No	No	No	Yes	No	EL	Yes
[56]	No	No	Yes	No	Yes	No	EL	Yes
[57]	No	No	Yes	No	Yes	Yes	EL	Yes
Our work	Yes	Yes	Yes	Yes	Yes	Yes	EL	Yes

Table 13: Comparisons of our model's advantages and disadvantages with the works in [55-57]

Researches	Approach	Advantages	Disadvantages
[55]	Examining the vector space model, an information retrieval technique and its variation	In this work, the authors have given an insider to the working of vector space model techniques used for efficient retrieval techniques. It is the bare fact that each system has its own strengths and weaknesses. What we have sorted out in the authors' work for vector space modeling is that the model is easy to understand and cheaper to implement, considering the fact that the system should be cost effective (i.e., should follow the space/time constraint. It is also very popular. Although the system has all these properties, it is facing some major drawbacks.	The drawbacks are that the system yields no theoretical findings. Weights associated with the vectors are very arbitrary, and this system is an independent system, thus requiring separate attention. Though it is a promising technique, the current level of success of the vector space model techniques used for information retrieval are not able to satisfy user needs and need extensive attention.
[56]	+Latent Dirichlet allocation (LDA). +Multi-label text classification tasks and apply various feature sets. +Several combinations of features, like bi-grams and uni-grams.	In this work, the authors consider multi-label text classification tasks and apply various feature sets. The authors consider a subset of multi-labeled files of the Reuters-21578 corpus. The authors use traditional TF-IDF values of the features and tried both considering and ignoring stop words. The authors also tried several combinations of features, like bi-grams and uni-grams. The authors also experimented with adding LDA results into vector space models as new features. These last experiments obtained the best results.	No mention
[57]	The K-Nearest Neighbors algorithm for English sentiment classification in the Cloudera	In this study, the authors introduce a new weighting method based on statistical estimation of the importance of a word for a specific categorization problem. One benefit of this method is that it can make feature selection	Despite positive results in some settings, GainRatio failed to show that supervised weighting methods are generally higher than unsupervised ones. The authors believe that ConfWeight is a promising supervised weighting technique that behaves gracefully

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	distributed system.	implicit, since useless features of the categorization problem considered get a very small weight. Extensive experiments reported in the work show that this new weighting method improves significantly the classification accuracy as measured on many categorization tasks.	both with and without feature selection. Therefore, the authors advocate its use in further experiments.
Our work	-We use the Self-training model based on the K-NN algorithm and the multi-dimensional vectors to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. -The sentiment lexicons of the bESD are based on the S6 coefficient (S6C) through the Google search engine with AND operator and OR operator. -The novel model uses the multi-dimensional vectors according to the sentiment lexicons. The advantages and disadvantages of the proposed model are shown in the Conclusion section.		

Table 14: Comparisons of our model with the latest sentiment classification models (or the latest sentiment classification methods) in [58-70]

Studies	S6C	CT	Sentiment Classification	PNS	SD	DT	Language	VSM
[58]	No	No	Yes	NM	Yes	Yes	Yes	vector
[59]	No	No	Yes	NM	Yes	Yes	NM	NM
[60]	No	No	Yes	NM	Yes	Yes	EL	NM
[61]	No	No	Yes	NM	Yes	Yes	NM	NM
[62]	No	No	Yes	No	No	No	EL	No
[63]	No	No	Yes	No	No	No	EL	No
Our work	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 15: Comparisons of our model's positives and negatives the latest sentiment classification models (or the latest sentiment classification methods) in [58-70]

Studies	Approach	Positives	Negatives
[58]	The Machine Learning Approaches Applied to Sentiment Analysis-Based Applications	The main emphasis of this survey is to discuss the research involved in applying machine learning methods, mostly for sentiment classification at document level. Machine learning-based approaches work in the following phases, which are discussed in detail in this work for sentiment classification: (1) feature extraction, (2) feature weighting schemes, (3) feature selection, and (4) machine-learning methods. This study also discusses the standard free benchmark datasets and evaluation methods for sentiment analysis. The authors conclude the research with a comparative study of some state-of-the-art methods for sentiment analysis and some possible future research directions in opinion mining and sentiment analysis.	No mention
[59]	Semantic Orientation-Based Approach for Sentiment Analysis	This approach initially mines sentiment-bearing terms from the unstructured text and further computes the polarity of the terms. Most of the sentiment-bearing terms are multi-word features unlike bag-of-words, e.g., "good movie," "nice cinematography," "nice actors," etc. Performance of semantic orientation-based approach has been limited in the literature due to inadequate coverage of multi-word features.	No mention
[60]	Exploiting New Sentiment-Based Meta-Level Features for Effective Sentiment Analysis	Experiments performed with a substantial number of datasets (nineteen) demonstrate that the effectiveness of the proposed sentiment-based meta-level features is not only superior to the traditional bag-of-words representation (by up to 16%) but also is also superior in most cases to state-of-art meta-level features previously proposed in the literature for text classification tasks that do not take into aS6Count any idiosyncrasies of sentiment analysis. The authors' proposal is also largely superior to the best lexicon-based methods as well as to supervised combinations of them. In fact, the proposed approach is	A line of future research would be to explore the authors' meta features with other classification algorithms and feature selection techniques in different sentiment analysis tasks such as scoring movies or products aS6Cording to their related reviews.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

		the only one to produce the best results in all tested datasets in all scenarios.	
[61]	Rule-Based Machine Learning Algorithms	The proposed approach is tested by experimenting with online books and political reviews and demonstrates the efficacy through Kappa measures, which have a higher accuracy of 97.4% and a lower error rate. The weighted average of different accuracy measures like Precision, Recall, and TP-Rate depicts higher efficiency rate and lower FP-Rate. Comparative experiments on various rule-based machine learning algorithms have been performed through a ten-fold cross validation training model for sentiment classification.	No mention
[62]	The Combination of Term-Counting Method and Enhanced Contextual Valence Shifters Method	The authors have explored different methods of improving the accuracy of sentiment classification. The sentiment orientation of a document can be positive (+), negative (-), or neutral (0). The authors combine five dictionaries into a new one with 21,137 entries. The new dictionary has many verbs, adverbs, phrases and idioms that were not in five dictionaries before. The study shows that the authors' proposed method based on the combination of Term-Counting method and Enhanced Contextual Valence Shifters method has improved the accuracy of sentiment classification. The combined method has accuracy 68.984% on the testing dataset, and 69.224% on the training dataset. All of these methods are implemented to classify the reviews based on our new dictionary and the Internet Movie Database data set.	No mention
[63]	Naive Bayes Model with N-GRAM Method, Negation Handling Method, Chi-Square Method and Good-Turing Discounting, etc.	The authors have explored the Naive Bayes model with N-GRAM method, Negation Handling method, Chi-Square method and Good-Turing Discounting by selecting different thresholds of Good-Turing Discounting method and different minimum frequencies of Chi-Square method to improve the accuracy of sentiment classification.	No Mention
Our work	-We use the Self-training model based on the K-NN algorithm and the multi-dimensional vectors to classify one document of the testing data set into either the positive polarity or the negative polarity in both the sequential environment and the distributed system. -The sentiment lexicons of the bESD are based on the S6 coefficient (S6C) through the Google search engine with AND operator and OR operator. -The novel model uses the multi-dimensional vectors according to the sentiment lexicons. The positives and negatives of the proposed model are given in the Conclusion section.		

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

APPENDIX OF CODES

ALGORITHM 1: performing 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: transferring one English document into one multi-dimensional vector in the sequential environment.
ALGORITHM 5: transferring all the documents of the testing data set into the multi-dimensional vectors in the sequential environment.
ALGORITHM 6: transferring all the positive documents of the training data set into all the multi-dimensional vectors, called the positive group of the training data set in the sequential system
ALGORITHM 7: transferring all the negative documents of the training data set into all the multi-dimensional vectors, called the negative group of the training data set in the sequential environment
ALGORITHM 8: performing the Hadoop Map phase of transferring each English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera
ALGORITHM 9: performing the Hadoop Reduce phase of transferring each English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera
ALGORITHM 10: implementing the Hadoop Map phase of the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system
ALGORITHM 11: implementing the Hadoop Reduce phase of the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system
ALGORITHM 12: implementing the Hadoop Map phase of transferring one document into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system.
ALGORITHM 13: implementing the Hadoop Reduce phase of transferring one document into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system
ALGORITHM 14: performing the Hadoop Map phase of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.
ALGORITHM 15: implementing the Hadoop Reduce phase of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.
ALGORITHM 16: performing the Hadoop Map phase of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.
ALGORITHM 17: implementing the Hadoop Reduce phase of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.
ALGORITHM 18: classifying all the documents of the testing data set into either the positive or the negative in the sequential system by using the self-training model with the K-NN algorithm and the S6C
ALGORITHM 19: performing the Hadoop Map phase of classifying all the documents of the testing data set into either the positive or the negative in the sequential system by using the self-training model with the K-NN algorithm and the S6C
ALGORITHM 20: implementing the Hadoop Reduce phase of classifying all the documents of the testing data set into either the positive or the negative in the sequential system by using the self-training model with the K-NN algorithm and the S6C.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ALGORITHM 1: performing 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: Each term in the 55,000 terms, do repeat:
Step 2: By using eq. (5), eq. (6), and eq. (7) of the calculating a valence of one word (or one phrase) in English in the section (4.1.1), the sentiment score and the polarity of this term are identified. The valence and the polarity are calculated by using the S6C 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;
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. (5), eq. (6), and eq. (7) of the calculating a valence of one word (or one phrase) in English in the section (4.1.1), the sentiment score and the polarity of this term are identified. The valence and the polarity are calculated by using the S6C 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: Add this term into the basis English sentiment dictionary (bESD);
Step 2: Return bESD;
End;

ALGORITHM 4: transferring one English document into one multi-dimensional vector in the sequential environment.

Input: one English document
Output: the multi-dimensional vector
Begin
Step 1: Split the English document into many separate sentences based on "." Or "!" or "?";
Step 2: Set Multi-dimensionalVector := { } { } with n_max rows and m_max columns;
Step 3: Set i := 0;
Step 4: Each sentence in the sentences of this document, do repeat:
Step 5: Multi-dimensionalVector[i][] := {};
Step 6: Set j := 0;
Step 7: Split this sentence into the meaningful terms (meaningful words or meaningful phrases);
Step 8: Get the valence of this term based on the sentiment lexicons of the bESD;
Step 9: Add this term into Multi-dimensionalVector[i];
Step 10: Set j := j+ 1;
Step 11: End Repeat – End Step 4;
Step 12: While j is less than m_max, repeat:
Step 13: Add {0} into Multi-dimensionalVector[i];
Step 14: Set j := j+1;
Step 15: End Repeat – End Step 12;
Step 16: Set i := i+1;
Step 17: End Repeat – End Step 4;
Step 18: While i is less than n_max, repeat:
Step 19: Add the vector {0} into Multi-dimensionalVector;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 20: Set i := i+1;
Step 21: End Repeat – End Step 18;
Step 22: Return Multi-dimensionalVector;
End;

ALGORITHM 5: transferring all the documents of the testing data set into the multi-dimensional vectors in the sequential environment.

Input: the documents of the testing data set
Output: the multi-dimensional vectors of the testing data set
Begin
Step 1: Set TheMulti-dimensionalVectors := {}
Step 2: Each document in the documents of the testing data set, do repeat:
Step 3: OneMulti-dimensionalVector := the algorithm 4 to transfer one English document into one multi-dimensional vector in the sequential environment with the input is this document;
Step 4: Add OneMulti-dimensionalVector into TheMulti-dimensionalVectors;
Step 5: End Repeat- End Step 2;
Step 6: Return TheMulti-dimensionalVectors;
End;

ALGORITHM 6: transferring all the positive documents of the training data set into all the multi-dimensional vectors, called the positive group of the training data set in the sequential system

Input: all the positive documents of the training data set;
Output: the positive multi-dimensional vectors, called the positive group
Begin
Step 1: Set ThePositiveMulti-dimensionalVectors := null;
Step 2: Each document in the positive documents, repeat:
Step 3: Multi-dimensionalVector := the algorithm 4 to transfer one English document into one multi-dimensional vector in the sequential environment with the input is this document;
Step 4: Add Multi-dimensionalVector into ThePositiveMulti-dimensionalVectors;
Step 5: End Repeat – End Step 2;
Step 6: Return ThePositiveMulti-dimensionalVectors;
End;

ALGORITHM 7: transferring all the negative sentences of the training data set into all the one-dimensional vectors, called the negative group of the training data set in the sequential environment

Input: all the negative sentences of the training data set;
Output: the negative multi-dimensional vectors, called the negative vector group
Begin
Step 1: Set TheNegativeMulti-dimensionalVectors := null;
Step 2: Each document in the negative documents, repeat:
Step 3: Multi-dimensionalVector := the algorithm 4 to transfer one English document into one multi-dimensional vector in the sequential environment with the input is this document;
Step 4: Add Multi-dimensionalVector into TheNegativeMulti-dimensionalVectors;
Step 5: End Repeat – End Step 2;
Step 6: Return TheNegativeMulti-dimensionalVectors;
End;

ALGORITHM 8: performing the Hadoop Map phase of transferring each English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera

Input: one sentence and the bESD;
Output: one term (one meaningful word/or one meaningful phrase) which the valence is identified
Begin
Step 1: Input this sentence and the bESD into the Hadoop Map in the Cloudera system;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 2: Split this sentence into the many meaningful terms (meaningful words/or meaningful phrases) based on the bESD;
Step 3: Each term in the terms, do repeat:
Step 4: Identify the valence of this term based on the bESD;
Step 5: Return this term; //the output of the Hadoop Map phase.
End;

ALGORITHM 9: performing the Hadoop Reduce phase of transferring each English sentence into one one-dimensional vector based on the sentiment lexicons of the bESD in Cloudera

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

ALGORITHM 10: implementing the Hadoop Map phase of the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system

Input: one document
Output: one one-dimensional vector
Begin
Step 1: Input this document into the Hadoop Map in the Cloudera system.
Step 2: Split this document into the sentences;
Step 3: Each sentence in the sentences, do repeat:
Step 4: One-dimensionalVector := null;
Step 5: Split this sentence into the meaningful terms;
Step 6: Each term in the meaningful terms, repeat:
Step 7: Get the valence of this term based on the sentiment lexicons of the bESD;
Step 8: Add this term into One-dimensionalVector;
Step 9 End Repeat – End Step 6;
Step 10: Return this One-dimensionalVector;
Step 11: The output of the Hadoop Map is this One-dimensionalVector;
End;

ALGORITHM 11: implementing the Hadoop Reduce phase of the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system

Input: One-dimensionalVector - one one-dimensional vector of the Hadoop Map (the input of the Hadoop Reduce is the output of the Hadoop Map)
Output: the multi-dimensional vector of the English document – Multi-dimensionalVector;
Begin
Step 1: Receive One-dimensionalVector;
Step 2: Add this One-dimensionalVector into One-dimensionalVector;
Step 3: Return Multi-dimensionalVector;
End;

ALGORITHM 12: implementing the Hadoop Map phase of transferring one document into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system.

Input: the documents of the testing data set
Output: one multi-dimensional vector (corresponding to one document)
Begin
Step 1: Input the documents of the testing data set into the Hadoop Map in the Cloudera system.
Step 3: Each document in the documents of the testing data set, do repeat:
Step 4: the multi-dimensional vector := the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system in Figure 7 with the input is this document;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 5: Return this multi-dimensional vector;
Step 6: The output of the Hadoop Map is this multi-dimensional vector;
End;

ALGORITHM 13: implementing the Hadoop Reduce phase of transferring one document into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system

Input: one multi-dimensional vector of the Hadoop Map (the input of the Hadoop Reduce is the output of the Hadoop Map)
Output: the multi-dimensional vectors of the English documents of the testing data set
Begin
Step 1: Receive one multi-dimensional vector of the Hadoop Map
Step 2: Add this multi-dimensional vector into the multi-dimensional vectors of the testing data set;
Step 3: Return the multi-dimensional vectors of the testing data set;
End;

ALGORITHM 14: performing the Hadoop Map phase of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

Input: the positive documents of the training data set
Output: one multi-dimensional vector (corresponding to one document of the positive documents of the training data set)
Begin
Step 1: Input the positive documents into the Hadoop Map in the Cloudera system.
Step 2: Each document in the documents, do repeat:
Step 3: MultiDimentionalVector := the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system in Figure 7
Step 4: Return MultiDimentionalVector ;
End;

ALGORITHM 15: implementing the Hadoop Reduce phase of transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

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

ALGORITHM 16: performing the Hadoop Map phase of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

Input: the negative documents of the training data set
Output: one multi-dimensional vector (corresponding to one document of the negative documents of the training data set)
Begin
Step 1: Input the negative documents into the Hadoop Map in the Cloudera system.
Step 2: Each document in the documents, do repeat:
Step 3: MultiDimentionalVector := the transferring one document into one multi-dimensional vector based on the sentiment lexicons of the bESD in the parallel system in Figure 7
Step 4: Return MultiDimentionalVector ;
End;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ALGORITHM 17: implementing the Hadoop Reduce phase of transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative group of the training data set) based on the sentiment lexicons of the bESD in the distributed system.

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

ALGORITHM 18: classifying all the documents of the testing data set into either the positive or the negative in the sequential system by using the self-training model with the K-NN algorithm and the S6C

Input: the documents of the testing data set and the training data set
Output: the results of the sentiment classification of the documents of the testing data set (positive, negative, or neutral);
Begin
Step 1: the creating a basis English sentiment dictionary (bESD) in a sequential environment (4.1.2);
Step 2: the algorithm 6 to transfer all the positive documents of the training data set into all the multi-dimensional vectors based on the sentiment lexicons of the bESD, called the positive group of the training data set in the sequential system
Step 3: the algorithm 7 to transfer all the negative documents of the training data set into all the multi-dimensional vectors based on the sentiment lexicons of the bESD, called the negative group of the training data set in the sequential environment.
Step 4: the algorithm 5 to transfer all the documents of the testing data set into the multi-dimensional vectors based on the sentiment lexicons of the bESD in the sequential environment.
Step 5: L : a data set of the labeled documents of the testing data set as follows: the positive group and the negative group of the training data set
Step 6: U : a set of the un-labeled documents of the training data set as follows:
Step 7: While until U = ∅
Step 7.1: Begin: Training a classifier h on L and using h to classify the data in U: Using the K-NN with the input of the K-NN is the multi-dimensional vectors of the testing data set, the positive group and the negative group of the training data set;

Step 7.2: Choose the 100 multi-dimensional vectors of the testing data set which are the best results of Step 7.1.
Step 7.3: U := U - the 100 multi-dimensional vectors of the testing data set;
Step 7.4: Add the 100 multi-dimensional vectors of the testing data set into either the positive group or the negative group of the training data set as follows: L := L + the 100 multi-dimensional vectors of the testing data set
Step 7.5: End While – End Step 7;
Step 8: Return the results of the sentiment classification of the documents of the testing data set (positive, negative, or neutral);
End;

ALGORITHM 19: performing the Hadoop Map phase of classifying all the documents of the testing data set into either the positive or the negative in the sequential system by using the self-training model with the K-NN algorithm and the S6C

Input: the documents of the testing data set and the training data set
Output: the results of the sentiment classification of 100 documents of the testing data set (positive, negative, or neutral); //the output of the Hadoop Map in the Cloudera system;
Begin
Step 1: the creating a basis English sentiment dictionary (bESD) in a distributed system (4.1.3)
Step 2: the transferring the documents of the testing data set into the multi-dimensional vectors of the document based on the sentiment lexicons of the bESD in the parallel system in Figure 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 3: the transferring the positive documents of the training data set into the positive multi-dimensional vectors (called the positive group of the training data set) based on the sentiment lexicons of the bESD in the distributed system in Figure 9
Step 4: the transferring the negative documents of the training data set into the negative multi-dimensional vectors (called the negative group of the training data set) based on the sentiment lexicons of the bESD in the distributed system in Figure 10
Step 5: Input the multi-dimensional vectors of the testing data set, the positive group and the negative group of the training data set into the Hadoop Map in the Cloudera system;
Step 6: L : a data set of the labeled documents of the testing data set as follows: the positive group and the negative group of the training data set
Step 7: U : a set of the un-labeled documents of the training data set as follows: the multi-dimensional vectors of the testing data set;
Step 8: While until U = ∅
Step 8.1: Begin: Training a classifier h on L and using h to classify the data in U: Using the K-NN with the input of the K-NN is the multi-dimensional vectors of the testing data set, the positive group and the negative group of the training data set;
Step 8.2: Choose the 100 multi-dimensional vectors of the testing data set which are the best results of Step 8.1.
Step 8.3: U := U - the 100 multi-dimensional vectors of the testing data set;
Step 8.4: Add the 100 multi-dimensional vectors of the testing data set into either the positive group or the negative group of the training data set as follows: L := L + the 100 multi-dimensional vectors of the testing data set
Step 8.5: Return the 100 multi-dimensional vectors of the testing data set;//the output of the Hadoop Map in the Cloudera system;
End;

ALGORITHM 20: implementing the Hadoop Reduce phase of classifying all the documents of the testing data set into either the positive or the negative in the sequential system by using the self-training model with the K-NN algorithm and the S6C.

Input: the 100 multi-dimensional vectors of the testing data set;//the output of the Hadoop Map in the Cloudera system;
Output: the results of the sentiment classification of the documents of the testing data set (positive, negative, or neutral);
Begin
Step 1: Receive the 100 multi-dimensional vectors of the testing data set;//the output of the Hadoop Map in the Cloudera system;
Step 2: Add the 100 multi-dimensional vectors into the results of the sentiment classification of the documents of the testing data set;
Step 3: Return the results of the sentiment classification of the documents of the testing data set;
End;