EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

Concept-based one-class SVM classifier with supervised term weighting scheme for imbalanced sentiment classification

Khanista Namee[1)] and Jantima Polpinij*[2)]

[1)]Faculty of Industrial Technology and Management, King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand
[2)]Intellect Laboratory, Faculty of Informatics, Mahasarakham University, Mahasarakham 44150, Thailand

Received 21 December 2020
Revised 11 February 2021
Accepted 8 March 2021

Abstract

Imbalanced sentiment is one of the key classification issues. Many studies have proposed imbalanced sentiment classification improvements, but the topic remains problematic as a major challenge. This paper proposes a method, called "*concept-based one-class SVM classifier*", to address imbalanced sentiment classification that consists of three main techniques. First, we apply Word2Vec and PageRank algorithms to extract "*concepts*" and their related terms (called "*members*") embedded in texts. The corpus of "*concepts*" is then used to prepare the dataset by replacing words with the "*concepts*". This reduces term ambiguity and also the size of word vectors. Second, supervised term weighting (STW) schemes are applied to determine the importance of a word in a document of a specific class. This reflects the class distinguishing power of each term. Finally, the one-class support vector machine (SVM) algorithm is used for sentiment classifier modeling. This has proved useful for imbalanced data classification, especially when the minority class lacks structure and is predominantly composed of small disjuncts or outliers. By combining these techniques, our proposed method may be able to competently identify and distinguish between the characteristics of each class, especially in the context of an imbalanced data scenario. After validating the proposed method with the hotel review dataset, and running experiments with different imbalanced ratios, our proposed method returned satisfactory results of recall, precision, and F1. We then selected the best model generated from our method and compared the results to the state-of-the-art method. Our proposed method returned better results than the state-of-the-art method, with improved scores of F1 at 3.19%. Moreover, if considering for the computational processing time, our proposed method is faster than the state-of-the-art method.

Keywords: Imbalanced sentiment classification, Word2Vec, Page rank, One-class SVM, Concept-based method, Supervised term weighting, Hotel reviews

1. Introduction

The growth and popularization of review sites, forums, blogs, and social media has mushroomed and the content available on social networking sites is now enormous. Manual analysis of customer comments is a time-consuming and error-prone task [1]. The need for automatic extraction and analysis of data from social websites has led to a study called "*sentiment analysis*", also known as "*opinion mining*" [2, 3]. It employs as the use of natural language processing (NLP), text mining, computational linguistics, and other methods to identify, extract, quantify, and study affective states and subjective information. Sentiment analysis provides valuable insights and assists organizations to equation effective business strategies by identifying both their strengths and weaknesses [2, 3]. Today, sentiment analysis is widely applied as a key component in many business applications e.g. customer relationship management (CRM) systems or other customer service applications [4].

In general, text classification is a common technique applied for sentiment analysis. This is adopted to analyze an incoming message and decides whether the underlying sentiment is positive, neutral, or negative. The task of applying text classification to sentiment analysis is called "*sentiment classification*" [5, 6]. This task is also helpful in business intelligence applications and recommender systems and allows user inputs and feedbacks to be quickly and accurately summarized.

To the best of our knowledge, there are many inherent issues in sentiment classification. A problem, called "*imbalanced sentiment classification*" has been confirmed that it is one of the key issues in sentiment classification [7-9]. This is because most existing studies assumed that numbers of samples in negative and positive classes could be balanced. This is not true in the real world. Imbalanced class distributions result when the sample number of one class in the training data is higher number than the other class. Simply speaking, the class with more samples is called the "*majority class (MA)*". Meanwhile, the class with fewer samples is called the "*minority class (MI)*". Although many methods, such as re-sampling [10, 11], one-class classification [12, 13], cost-sensitive learning [14], ensemble learning [10, 11], hybrid method [11], and deep learning [15, 16] have been studied and proposed to address imbalanced class distribution issues, this issue is not yet completely addressed. This is because classification algorithms cannot perform well on

*Corresponding author. Tel.: +668 8571 3155
Email address: jantima.p@msu.ac.th
doi: 10.14456/easr.2021.62

imbalanced datasets. Consequently, those classification algorithms cannot train a robust classifier model, especially when using with a deep learning method. Therefore, imbalanced sentiment classification remains a challenge for classification tasks.

In the last decade, there are the other approaches applied for handling of imbalanced text classification. First, it is to use supervised term weighting (STW) schemes because since these term weighting schemes may help to select appropriate features for use in text classification. Furthermore, the STW schemes can identify and distinguish between the characteristics of each class, since they may help to define the contribution of each term to a specific class [17-20]. However, although those studies presented good text classification results, almost all the work in this task has been performed using standard datasets (e.g. Reuters-21578). In addition to data volume increases becoming prohibitive for existing methods, the character of the issue can make some additional difficulties. Large data imbalances can expose from different specific areas such as medical data, social networks, and so on [21, 22]. The second solution proposed for improving of imbalanced text classification is to apply word embedding to handle imbalanced data sentiment analysis. Then, this solution also returned a better performance for imbalanced datasets [16]. However, although these solutions can help improving the performance of text classification with imbalanced data, they focused only on one contribution. In our perspective, if multiple contributions are applied concurrently, this may return better result of imbalanced sentiment classification. Consequently, resolving the imbalanced sentiment classification based on multiple contributions is a major challenge in this study.

This study proposes a method, called "*concept-based one-class SVM classifier*", to address imbalanced sentiment classification that consists of three main techniques. First, we apply Word2vec and PageRank algorithms to extract "*concepts*" and their related terms (called "*members*") embedded in texts. The corpus of "*concepts*" is then used to prepare the dataset by replacing words with the "*concepts*". This reduces term ambiguity and also the size of word vectors. Second, supervised term weighting (STW) schemes are applied to determine the importance of a word in a document of a specific class. This reflects the class distinguishing power of each term. Finally, the one-class SVM algorithm is used for sentiment classifier modeling. This algorithm has proved that it is useful for imbalanced data classification [12, 13], especially when the minority class lacks structure and is generally composed of outliers and disjuncts. The effectiveness of this proposed method is validated by on the dataset downloaded from TripAdvisor and Booking websites. Furthermore, the best model of our proposed method is also selected and compared with the state-of-the-art method proposed by Li et al. [23].

The rest of the paper is organized as follows. In Section 2, it describes our dataset. Section 3 explains the proposed method. Section 4 shows the experimental results and comparison results along with discussion. Finally, conclusion is presented in Section 5.

2. The dataset

The dataset is customer reviews relating hotel. They were gathered from TripAdvisor and Booking websites, downloaded during February 2020. They were formatted as XML files and differed in length. The original dataset consisted of 400,000 reviews. 200,000 customer reviews labeled as positive and the rest as negative. For Booking website, the customer reviews rated as 7, 8, 9, and 10 were assigned as the positive class, while those rated as 1, 2, 3 and 4 were assigned to the negative class. For TripAdvisor website, the customer reviews rated as 4 and 5 were assigned as the positive class, while those rated as 1 and 2 were assigned to the negative class. The customer reviews were written in English and their minimal length should be 10 words. Some examples are shown in Figure 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
   <reviewID>00002
      <from> TripAdvisor </from>
      <review> Staff was unfriendly and room is unclean and uncomfortable </review>
      <sentPolarity> Negative </sentPolarity >
   </reviewID>
</note>
```

Figure 1 Example of hotel review

In our experiments, we randomly select customer reviews for three subsets from the original dataset. The training sets and test sets are different. Also, 10-fold cross validation are applied when developing the sentiment classifiers. Finally, after obtaining the most appropriate sentiment classifiers, the test set is used to evaluate those classifiers. They can be summarized as Table 1.

Table 1 Summary of three subsets of data used in our experiments for imbalanced sentiment classification

Sub set No.	Training set		Test set	
	Total number of customer reviews in positive class	**Total number of customer reviews in Negative class**	**Total number of customer reviews in positive class**	**Total number of customer reviews in Negative class**
1	100,000	10,000	20,000	20,000
2	100,000	20,000	20,000	20,000
3	100,000	30,000	20,000	20,000

3. The proposed method

This section describes for the proposed method. It consists of two main steps -- namely generating concepts and its members by Word2Vec and PageRank algorithms and concept-based sentiment classification. The overview of the proposed method can be shown as Figure 2.

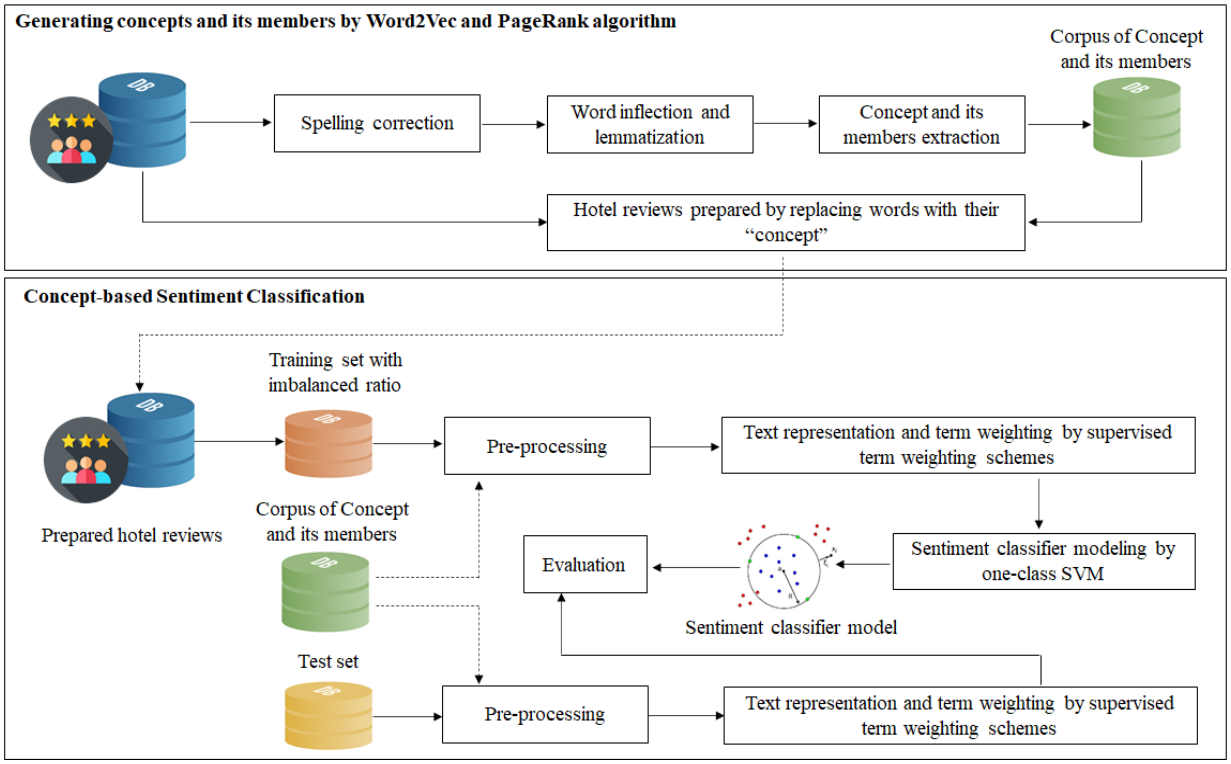


Figure 2 Overview of the proposed method

3.1 Generating concepts and its members by Word2Vec and PageRank algorithms

This section is the process of finding similar words as near neighbors in the vector space of hotel reviews. The most significant words are considered as representatives of the word vector, called "*concept*". Other words found in the vector and related to "*concept*" are called "*members*". This task consists of three processing steps that can be detailed as follows.

3.1.1 Spelling correction

Hotel reviews written by customers contain unstructured text that may contain spelling mistakes. Therefore, it is necessary to correct these spelling errors before processing to the next stage. In this step, some word forms are changed from short forms used in everyday speech and informal writing into longer forms. Words written as "*didn't*", "*don't*", and "'*s*" are changed to "*did not*", "*do not*", and "*is*", respectively. An example of spelling correction can be shown in Table 2.

Table 2 An example of spelling correction, word inflection, and lemmatization

Original hotel review	This is a beautiful hotel in an excellent location in Bangkok. The rooms are clean. The staff attended to our every need.
After spelling correction, word reflection, and lemmatization	This be a beautiful hotel in an excellent location in Bangkok. The room be clean. The staff attend to our every need.

3.1.2 Word inflection and lemmatization

Word inflection converts a word into singular form, where inflections take a variety of word forms and make an ambiguity during automatic language processing. While lemmatization removes the inflectional endings of a word such as comparative and superlative terms to simpler forms. Both word inflection and lemmatization are performed to avoid repetition and return to the base dictionary form of the word, known as the "*lemma*". An example of word inflection and lemmatization can also be shown in Table 2.

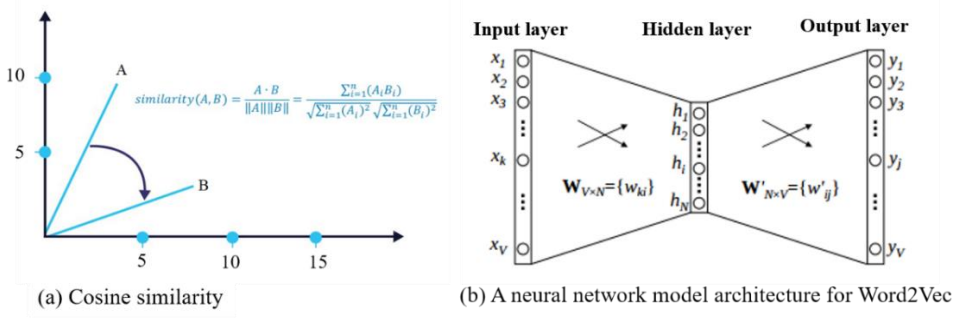
Finally, let w be significant words in a hotel review and sentiment-class be class label of that hotel review. A pre-processed hotel review *hr* can be represented as:

$hr = <w_1, w_2, w_3, \ldots, w_n, \text{"sentiment-class"}>$

Concept and its members extraction by Word2vec and PageRank algorithms. The most common representation of distributional semantics is called one-hot encoding, in which categorical variables are represented as binary vectors. Elements of this vector space representation consist of 0 and 1. However, this representation has some disadvantages. This is because it is difficult to make deductions about word similarity since large dimension can cause overfitting, while it is also computationally expensive. Therefore, word embeddings have been proposed and applied to capture attributional similarities between vocabulary items. Words that appear in similar contexts should be close or related to each other in the projected vector space.

A well-known word embedding is Word2Vec [24-26]. It is a predictive deep learning-based model that is applied to analyze words embedded in texts. Word2Vec attempts to learn relationships between words given a large corpus of texts and identifies similar words

by finding near neighbors in the vector space with cosine similarity, using a neural network to learn vector representation. The main mechanisms used in Word2Vec are shown in Figure 3. Figure 3 (b) represents a neural network model architecture for Word2Vec that is used to transform words into word vectors in an n-dimensional vector space, also called embedding, while Figure 3 (a) represents the cosine similarity used to calculate the distance between two vectors in an n-dimensional vector space. The distance apart represents how closely the words are related to each other.



(a) Cosine similarity (b) A neural network model architecture for Word2Vec

Figure 3 Main mechanisms used in Word2Vec

In this study, Word2Vec is applied to decide a threshold of counts of words. Words that appear only once or twice in a large corpus are probably uninteresting typos and garbage. There is not enough data to make any meaningful training on these words and they are best deleted. A reasonable value for minimum word counts is between 0 and 50 depending on the size of the corpora. Another critical parameter for the Word2Vec model is the dimension of the vectors that changes between 100 and 300. We use the maximum dimension of vector as 300 because dimensions larger than 300 require more training but this can lead to more accurate models. In this case, our dataset consists of 400,000 hotel reviews downloaded from TripAdvisor and Booking websites are used. This dataset consists of 10,165 word-vectors. However, named entities, stop-words, or infrequent words are not removed because the algorithm utilizes a sliding window to find vector representations. Simply speaking, nearby words are used to find vector representations.

The value of each word is calculated using the PageRank algorithm which performs with random walk [27]. The original PageRank algorithm used internet pages as nodes. However, in this study, the PageRank algorithm takes Word2Vec representations of words as nodes, and cosine similarity is applied to calculate edge weights between nodes. After PageRank values of words are calculated, words having the highest values can be considered as keywords of a text. The pseudocode of the PageRank algorithm used in this study can be shown as Figure 4.

Algorithm: PageRank

```
1.  Procedure PageRank(G, Iteration)          ; G: inlink file, iteration: number of iteration
2.      d ← 0.85                              ; daming factor: 0.85
3.      oh ← G                                ; get outlink count hash from G
4.      ih ← G                                ; get inlink hash from G
5.      N ← G                                 ; get number of nodes from G
6.      for all p in the graph do
7.          opg[p] ← 1/N                      ; initial PageRank
8.      end for
9.      while iteration > 0 do
10.         dp ← 0
11.         for all p that has no out-links do
12.             dp ← dp + d × (opg[p]/N)      ; get PageRank from nodes without out-links
13.         end for
14.         for all p in the graph do
15.             npg[p] ← dp + (1-d/N)         ; get PageRank from random jump
16.             for all ip in ih[p] do
17.                 npg[p] ← npg[p] + (d×(opg[ip]/oh[ip])   ; get PageRank from in-links
18.             end for
19.         end for
20.         opg ← npg                         ; update PageRank
21.         iteration ← iteration-1
22.     end while
23. end procedure
```

Figure 4 Pseudocode of PageRank algorithm [27]

An example of the result after performing Word2Vec and PageRank algorithms can be illustrated as Figure 5. Later, after obtaining vector representations of words, the weight of each word in the vector is calculated because weighting determines the importance of a term for a corpus. In this study, we apply a global weighting scheme, called "*inverse document frequency* (*idf*)", to calculate and assign a weight for each term word. The equation of *idf* is

$$idf(t_i) = \log\left(1 + \frac{N}{df(t_i)}\right) \quad (1)$$

where N is the whole number of documents in the corpus, and $df(t_i)$ is the number of documents in the corpus containing term t_i.

We are only interested in global weight because we need to know the importance of a term for a corpus. If a word in a vector has the highest weight score, it is a representative of that vector, called "*concept*". Meanwhile, other words in the same group become "*members*" of that "*concept*". Finally, we obtain 876 concepts.

After obtaining the "*concepts*" and their "*members*", words in an original hotel review that are "*member*" words will be replaced with their "*concept*". After preforming Word2Vec, suppose we obtain three groups of words. The first group consists of the word '*good*', '*well*', '*perfect*', '*nice*', '*best*', '*lovely*', and '*awesome*'. The second group consists of the word '*bad*', '*poor*', '*worst*', '*horrible*', and '*terrible*'. The third group consists of '*close*', '*near*', '*beside*', and '*opposite*'. Suppose the word "*good*" is selected as the "*concept*" of that group. Therefore, all "*member*" words in this group should be replaced by the word "*good*". The hotel reviews that are performed with the corpus of concepts and its members are called "*prepared hotel review*". An example of prepared hotel review is shown as Table 3. This is to decrease the feature vector length and reduce the ambiguity of natural language. Decreasing the feature vector length may seem to balance data in each class. This idea could improve performance of text classification because it returned the satisfactory results [28].

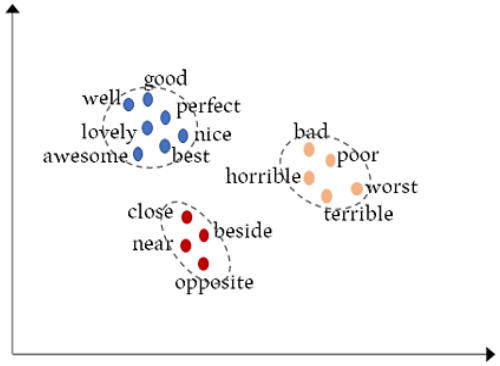


Figure 5 Examples of the results after performing Word2Vec and PageRank algorithms representing in 2D format

3.2 Concept-based sentiment classification

The proposed consists of three processing steps. The first processing step is to pre-process hotel reviews performed by NLP techniques namely word segmentation, entities identification and removal, and stop-word removal. Afterwards, the prepared reviews are represented in the format of the vector space model (VSM) and given their term weights by STW schemes. To obtain an appropriate term weighting scheme for imbalanced sentiment classification, this work studies for four term weighting schemes namely *term frequency – inverse document frequency* (*tf-idf*), *term frequency - relevance frequency* (*tf-rf*) [29], *term frequency - inverse gravity moment* (*tf-igm*) [30], and *term frequency - inverse document frequency of terms in classes -relevance frequency* (*tf-idfc-rf*) [31]. This stage is called text representation and term weighting. Finally, the effectiveness of this proposed method is validated by experiments of sentiment classification using one-class SVM classifiers.

Table 3 An example of prepared hotel review

Original hotel review	This is a good hotel in a nice location in Bangkok. The rooms are awesome. The staff is nice.
After spelling correction, word reflection, and lemmatization	This be a good hotel in a nice location in Bangkok. The room be awesome. The staff be good.
Hotel review after replacing words	This be a good hotel in a good location in Bangkok. The room be good. The staff be good.

3.2.1 Pre-processing

This study employs Natural Language Toolkit (NLTK) in Python [22] to apply NLP techniques. First, it is to tokenize text for determining the word boundaries in a text or a document by dividing the text into its component words. Later, proper nouns such as hotel or place names are identified and removed using a named entity recognition tool, called Stanford Named Entity Recognizer (NER) [32]. This is because these words are extrinsic and likely to be unnecessary for sentiment prediction of text. The final process of this stage is to remove stop-words (e.g. prepositions and articles) because they are useless and meaningless words that are unnecessary for sentiment prediction. Therefore, these words are filtered out from the hotel reviews.

3.2.2 Text representation and term weighting

After preprocessing, the hotel reviews are represented in VSM format. The size of the vector is equal to the number of concepts mentioned in Section 3.1. This study uses both of unsupervised term weighting (UTW) and supervised term weighting (STW) schemes.

1) Unsupervised term weighting

This study also uses the well-known UTW scheme, called *tf-idf*, for giving term weight. This term weighting scheme has been demonstrated that it may be effective for information retrieval (IR). However, it might be unsuitable for text classification task. This is because *tf-idf* ignores the class labels of training documents. Furthermore, it may assign large weights to the words, but lacking the discrimination power of text [30]. This study also uses this term weight scheme because it is often used in a traditional method of sentiment classifier modelling. Then, the equation of *tf-idf* is shown as follows:

$$tf - idf_{t,d} = tf_{t,d} \times idf_t \tag{2}$$

The *tf* equation can be:

$$tf_{t,d} = log(1 + f_{t,d}) \tag{3}$$

where $f_{t,d}$refers to the number of times that a particular term *t* appears in document d, while idf_t can be presented as (1).

2) Supervised term weighting

Many researchers have proposed and used STW schemes for text classification because these offer more effective performance returns. Most STW schemes consider term distribution in the classes of interest, while weighting a term helps to improve its discriminating powers for text classification tasks [20, 30, 31, 33]. As a result, STW schemes can help to improve the performance of sentiment analysis and other text mining tasks [20, 30, 31, 33]. Consequently, we also used STW schemes in this study. As STW schemes have a competence of identifying and distinguishing between the characteristics of each class, any opinion lexicon resource, i.e. SentiWordNet, is unnecessary here.

To obtain an appropriate term weighting scheme for imbalanced sentiment classification, this work studies for 3 term weighting schemes namely *tf-rf* [29], *tf-igm* [30], *tf-idfc-rf* [31]. Before giving detail of each STW scheme, it needs to understand the concept of STW. STW schemes weight terms by exploiting the known class information in training corpus. The fundamental elements of STW are depicted in Table 4.

Table 4 The basic elements of supervised term weighting

	C_k	$\bar{C}_k$
t_i	*A*	*C*
$\bar{t}_i$	*B*	*D*

In Table 4, the importance of a term t_i for a class c_k is represented as follows. *A* is the total number of documents in class c_k where the term t_i appears at least once, while *B* is the total number of documents belonging to class c_k where the term t_i does not appear. *C* is the total number of documents not belonging to class c_k, where the term t_i appears at least once. *D* is the number of documents not belonging to class *ck*, where the term t_i does not appear. Meanwhile, *N* is the whole number of documents in the corpus, given equation as $N = A + B + C + D$. Np is the whole number of documents in positive class, where $Np = A + B$. N_n is the whole number of documents in the negative class, where $N_n = C + D$.

Considering the basic elements presented in Table 4. Each STW term weighting scheme is described as follows.

(1) $tf - rf$

The *tf-rf* considers term distributions in both positive and negative classes. However, only documents containing the *rf* of the terms are interested. The equation of *tf-rf* is represented in (4) and the minimal denominator should be 1 for avoiding division by zero as:

$$tf - rf(t_i) = tf(t_i, d_i) \times \log\left(2 + \frac{A}{\max(1, C)}\right) \tag{4}$$

(2) tf-igm

The *tf-igm* is proposed to improve preciseness of measurement for the class distinguishing power of a term. Then, *inverse gravity moment* (*igm*) is the main mechanism of this STW. Then, *igm* is an application of the "*gravity moment (gm)*" concept from the physics. The *tf-igm* is to combine *term frequency* (*tf*) with the *igm* measure. The equation of *tf-igm* is represented in (5).

$$tf - igm(t_i) = tf(t_i, d_i) \times \left(1 + \lambda \times igm(t_i)\right) \tag{5}$$

Meanwhile, the *igm* is indicated in (4), where f_{ir} (r = 1, 2, ..., M) indicates the total number of documents that contain the term t_i in the *r*-th class. These documents are sorted in descending order. Thus, f_{i1} represents the frequency of t_i in the class in which it appears most often. The equation of *igm* is represented in (6).

$$igm(t_i) = \frac{f_{i1}}{\sum_{r=1}^{M} f_{ir} \times r} \tag{6}$$

In (5), let λ be an adjustable coefficient used to maintain the relative balance between the global factor *igm* and local factor *tf* in the weight of a term. The λ coefficient has a default value of 7.0 and be set as a value between 5.0 and 9.0 [30].

(3) tf-idfc-rf

The *tf-idfc-rf* was proposed by modifying the *idf* concept. It is also inspired in *tf-rf* since it calculates the *rf* of a term. To avoid division by zero, it adjusts the denominators with (A + 1) for *idfc* and (C + 1) for *rf* as mentioned in [29]. However, the *rf* component is adjusted the numerator with (A + 1) to avoid *log*(0). The equation of *tf-idfc-rf* is represented in

$$sqrt\_tf - idfc - rf(t_i) = SQRT\_TF(t_i, d_i) \times \log_2\left(\frac{2+\max(A,C)}{\max(2,\min(A,C))}\right) \times \sqrt{B+D} \quad (7)$$

3.2.3 Sentiment classifier modeling

To model sentiment classifiers with imbalanced data, we apply one-class SVM because this algorithm has been confirmed by some studies as a solution to address the class imbalance problem [12, 13]. The concept behind the one-class SVM is to identify outliers in a dataset, called "*anomaly detection*", while the rest is called "*normal*".

In this study, one-class SVM is used for binary imbalanced classification problems. The negative case is regarded as "*normal*", while the positive case is considered as an "*anomaly*". We apply the one-class SVM algorithm proposed by [12] because this algorithm is a sophisticated method that uses artificially generated outliers to optimize one-class SVM parameters and balance over-fitting and under-fitting. This algorithm is called Support Vector Data Description (SVDD) and relies on identifying the smallest hypersphere (with radius *r* and center *c*) consisting of all the data points. Formally, the problem can be defined in the following constrained optimization form,

$$\min_{r,c} r^2 \quad (8)$$

$$\text{Subject to } \|\phi(x_i) - c\|^2 \leq r^2 + \xi_i, \text{for all } i = 1, 2, \dots, l \quad (9)$$

However, the above formulation is highly restrictive and sensitive to the presence of outliers. Therefore, a flexible formulation that allows for the presence of outliers is presented

$$\min_{r,c} r^2 + \frac{1}{vl}\sum_{i=1}^{l} \xi_i \quad (10)$$

Subject to:

$$\|\phi(x_i) - c\|^2 \leq r^2 + \xi_i, \text{for all } i = 1, 2, \dots, l \quad (11)$$

Using the optimal Karush-Kuhn-Tucker (KKT) conditions, this can be represented as:

$$c = \sum_{i=1}^{l} \alpha_i \phi(x_i) \quad (12)$$

where α_i is the solution to the following optimization problem:

$$\max_{\alpha} \sum_{i=1}^{l} \alpha_i k(x_i, x_j) - \sum_{i,j}^{l} \alpha_i \alpha_j k(x_i, x_j) \quad (13)$$

Subject to:

$$\sum_{i=1}^{l} \alpha_i = 1 \text{ and } 0 \leq \alpha_i \leq \frac{1}{vl} \text{ for all } i = 1, 2, \dots, l \quad (14)$$

The kernel function then provides additional flexibility to the one-class SVM algorithm. We chose the Gaussian Radial Base Function (RBF) kernel because some studies mentioned that this algorithm performs best when the Gaussian kernel is used.

4. Experimental results and discussion

4.1 The measurement techniques for evaluation

In this study, recall, precision, and F1 are measures to evaluate the performance of imbalanced sentiment classification. The recall is the relevant documents that are retrieved from the total relevant documents of a dataset, and the precision refers to the closeness of two or more measurements to each other. F1 provides a single score of harmonic mean that balances both the concerns of precision and recall in one number.

4.2 The experimental results of the proposed method

We verified the effectiveness of our proposed method by running experiments with different imbalanced ratios, where the imbalanced ratio is the number of samples in the majority class divided by those in the minority class. We used imbalanced ratios as 10:1, 10:2 and 10:3, respectively.

This section also compares our proposed method with the traditional method of sentiment classification. STW schemes are compared with the well-known *tf-idf* method to assess their performances for imbalanced sentiment analysis. The traditional method was driven on the use of words generated by the tokenization process, and then these words were represented by the format of VSM along with term weighting. Finally, the traditional method modeled the sentiment classifiers using one-class SVM algorithm.

Experimental results presented in Table 5 show that our proposed method improved the recall, precision and F1 over the traditional method and the proposed method with *tf-igm* as the term weighting returns the most appropriate results. There are three reasons for these results described as follows.

First, using "*concepts*" as features may help to lessen the impact of polysemous terms in textual hotel reviews by reducing the ambiguity and noise inherent in their bag-of-words representations [34]. When language ambiguity is reduced, the performance of sentiment classification increases.

Second, by giving weight to each term, STW schemes can competently identify and distinguish between the characteristics of each class because they consider term distribution in the classes of interest. Thus, these term weighting schemes improve the discriminating powers for text classification tasks [20, 30, 31, 33].

Table 5 The experimental results of the proposed method

Imbalance ratio (pos : neg)	Method	Term weighting scheme	Recall	Precision	F1
10:1	Traditional method with one-class SVM	*tf-idf*	0.670	0.513	0.581
		tf-rf	0.725	0.553	0.628
		tf-igm	0.750	0.588	0.659
		tf-idfc-rf	0.725	0.553	0.628
	Proposed method with one-class SVM	*tf-idf*	0.695	0.586	0.636
		tf-rf	0.745	0.633	0.685
		tf-igm	0.785	0.675	0.726
		tf-idfc-rf	0.745	0.633	0.685
10:2	Traditional method with one-class SVM	*tf-idf*	0.670	0.563	0.612
		tf-rf	0.725	0.603	0.658
		tf-igm	0.750	0.630	0.685
		tf-idfc-rf	0.725	0.603	0.658
	Proposed method with one-class SVM	*tf-idf*	0.695	0.621	0.656
		tf-rf	0.745	0.653	0.696
		tf-igm	0.785	0.691	0.735
		tf-idfc-rf	0.745	0.653	0.696
10:3	Traditional method with one-class SVM	*tf-idf*	0.670	0.605	0.636
		tf-rf	0.725	0.638	0.679
		tf-igm	0.750	0.670	0.708
		tf-idfc-rf	0.725	0.638	0.679
	Proposed method with one-class SVM	*tf-idf*	0.695	0.625	0.658
		tf-rf	0.745	0.694	0.719
		tf-igm	0.785	0.750	0.767
		tf-idfc-rf	0.745	0.694	0.719

Third, this study applied one-class SVM to create sentiment classifier models that returned satisfactory results. There are two reasons. First, the character of SVM is also good for small size data vectors. Consequently, it is unsurprised for this study because our word vector used in this study is quite small and limited to only 876 features. It depends on the number of concepts that are obtained by performing Word2Vec and PageRank algorithms (mentioned as Section 3.1). Second, the one-class SVM algorithm may be suitable for input documents from the imbalanced training set because this algorithm captured the characteristics of training documents and distinguished between them and the appearance of potential outliers. Therefore, it might be effective for imbalanced classification dataset. That is, the one-class SVM algorithm may have a power for identifying and distinguishing class although the dataset is unstructured form together with outliers and the minority class contains few documents.

However, the advantages of one-class SVM may come at a price of ignoring all available information about the majority class. Therefore, it may not be suitable for applying to some applications where information about the majority class is necessary for expectative results. Therefore, when using one-class SVM, it should be used carefully since it may not fit other specific applications [35, 36]. Finally, the best sentiment classifier models generated from our proposed method were selected and compared to the-state-of-the-art method proposed by Li et al. [23]. Then, they were compared in the same environment with the imbalanced ratio at 10:1, 10:2 and 10:3, respectively.

4.3 Comparison with the state-of-the-art method

In this section, we compare our proposed method with the method proposed by Li et al. [23], who introduced a sentiment classification method that synthesized both universal and domain-specific knowledge. Unlike previous studies that induced lexicon-based features from traditional sentiment lexicons, Li et al. [23] presented a label propagation method with glove-embedding to generate corpus-adaptive sentiment lexicons. While a domain-specific sentiment lexicon was induced by utilizing Singular Value Decomposition (SVD) embedding into the label propagation model. Later, the universal and domain-specific lexicons together with the two word-embeddings were combined to train an ensemble classifier. Moreover, Li et al. [23] proposed an over-sampling method to address the imbalanced data problem caused by the skewed distribution of sentiment polarity that often occurs in a domain-specific corpus.

We compared our proposed method with Li et al. [23] because both studies addressed similar ideas using lexicon-based features. We generated features as words embedded in texts by Word2Vec and PageRank algorithms, while Li et al. [23] generated features

from traditional sentiment lexicons. The state-of-the-art method was also compared with our study using the same experimental setup with our datasets. The results of the comparisons evaluating by F1 are shown in Table 6.

Results in Table 6 show that our proposed method improved the F1 score over the-state-of-the-art method, following the reasonings described in the previous section. Moreover, if considering for the computational processing time, our proposed method is faster than the state-of-the-art method. In summary, our proposed method may be suitable for the dataset used in this study. However, this method should be used carefully because it may not fit other specific applications.

Table 6 Comparison of the proposed algorithm with the-state-of-the-arts under imbalanced scenario

Method	Imbalance ratio (pos: neg)	F1	Average F1
Li et al. [23]	10:1	0.700	0.720
	10:2	0.720	
	10:3	0.740	
Proposed method	10:1	0.726	0.743
(*Concept-based one-class SVM classifier with tf-igm*)	10:2	0.735	
	10:3	0.767	
Improved average scores of F1			3.19%

5. Conclusions

Imbalanced class distribution occurs when the number of observations belonging to one class is significantly higher or lower than those belonging to other classes. Most classification tasks are not equipped to handle imbalanced classes and tend to show bias toward the majority this issue raises problems in machine learning sentiment classification, and predicting an outcome becomes difficult when there is not sufficient data to use as an effective training set. This key problematic issue in sentiment classification is called "*imbalanced sentiment classification*".

Many methods such as re-sampling, one-class classification, cost-sensitive learning, ensemble learning, hybrid methods and deep learning have been proposed. However, this issue has not yet been completely addressed because these methods cannot perform well on imbalanced datasets, while classification algorithms cannot train a robust model. Furthermore, using fewer samples, referred to as the minority class, is not appropriate since deep learning requires large amounts of training data. Therefore, imbalanced sentiment classification remains a challenge for classification tasks, where previous studies have often focused on only one contribution. As a result, the proposed solutions cannot comprehensively cover all aspects of the issue. From our perspective, applying multiple contributions concurrently may return better results of imbalanced sentiment classification. Consequently, resolving the imbalanced sentiment classification problem based on multiple contributions is a major challenge.

In this study, we present a method, called "*concept-based one-class SVM classifier*" to address imbalanced sentiment classification. Our method consists of three main techniques. First, we consider that language ambiguity in texts affects the performance of imbalanced sentiment classification. This is because ambiguous terms may be difficult to identify and distinguish between the characteristics of each class. Therefore, we consider a "*concept*" as a keyword and other words related to that "*concept*" should be replaced by the "*concept*". The concepts and their related terms (called as "*members*") embedded in texts can be captured using the Word2Vec and PageRank algorithms. The corpus of concepts is used to prepare hotel datasets by replacing words with the "*concept*". This reduces term ambiguity and also the size of word vectors. Second, STW schemes are well-recognized to competently identify and distinguish between the characteristics of each class, especially when using an imbalanced data scenario. Thus, they are also used for improving the performance of imbalanced sentiment classification. Third, we consider that when modeling classifier models with one class, an algorithm can capture the density of the majority class, and classify documents on the extremes of the density function as outliers. This may help to separate classes clearly, although some classes contain scant data. The one-class SVM algorithm has proved useful for imbalanced data classification, especially when the minority class lacks any structure and is predominantly composed of small disjuncts or noisy instances.

After verifying the effectiveness of the proposed method, using the hotel review dataset, and running experiments with different imbalanced ratios, our proposed method returned satisfactory results of recall, precision, and F1.

Finally, we selected the best model generated from our method and compared the results to the state-of-the-art method. Our method returned better results than the state-of-the-art method, with improved scores of F1 at 3.19%. However, if considering for the computational processing time, our proposed method is faster than the state-of-the-art method.

6. Acknowledgements

This research was funded by King Mongkut's University of Technology North Bangkok. Contract no. KMUTNB-62-DRIVE-17.

7. References

[1] Wu Y, Wei F, Liu S, Au N, Cui W, Zhou H, et al. Opinion seer: interactive visualization of hotel customer feedback. IEEE Trans Visual Comput Graph. 2010;16:1109-18.
[2] Liu B, Zhang L. A survey of opinion mining and sentiment analysis. In: Aggarwal C, Zhai C, editors. Mining Text Data. New York: Springer; 2012. p. 415-63.
[3] Lakshmanaprabu SK, Shankar K, Gupta D, Khanna A, Rodrigues J, Pinheiro PR, et al. Ranking analysis for online customer reviews of products using opinion mining with clustering. Complexity. 2018;2018:1-9.
[4] Medhat W, Hassan A, Korashy H. Sentiment analysis algorithms and applications: a survey. Ain Shams Eng J. 2014;5:1093-113.
[5] Liu SM, Chen JH. A multi-label classification-based approach for sentiment classification. Processing. 2015;42:1083-93.
[6] Catal C, Nangir M. A sentiment classification model based on multiple classifiers. Appl Soft Comp. 2017;50:135-41.
[7] Li S, Zhou G, Wang Z, Lee SYM. Imbalanced sentiment classification. Proceedings of the 20th ACM Conference on Information and Knowledge Management; 2011 Oct 24-28; Glasgow, United Kingdom. New York: IEEE; 2011. p. 28281-90.

[8] Li Y, Guo H, Zhang Q, Gu M, Yang J. Imbalanced text sentiment classification using universal and domain-specific knowledge. Knowl Base Syst. 2018;160:1-15.
[9] Wu F, Wu C, Liu J. Imbalanced sentiment classification with multi-task learning. Proceedings of the 27th ACM international conference on information and knowledge management; 2018 Oct 22-26; Torino, Italy. New York: Association for Computing Machinery; 2018. p. 1631-4.
[10] Wang S, Minku LL, Yao X. Resampling-based ensemble methods for online class imbalance learning. IEEE Trans Knowl Data Eng. 2015;27:1356-68.
[11] Prusa J, Khoshgoftaar TM, Dittman DJ, Napolitano A. Using random under sampling to alleviate class imbalance on tweet sentiment data. 2015 IEEE International conference on information reuse and integration; 2015 Aug 13-15; San Francisco, USA. New York: IEEE; 2015. p. 197-202.
[12] Zhuang L, Dai H. Parameter estimation of one-class SVM on imbalance text classification. 19th Conference of the Canadian society for computational studies of intelligence; 2006 Jun 7-9; Quebec City, Canada. Berlin: Springer; 2006. p. 538-49.
[13] Klikowski J, Wozniak M. Employing one-class SVM classifier ensemble for imbalanced data stream classification. International conference on computational science; 2020 Jun 3-5; Amsterdam, Netherlands. Berlin: Springer; 2020. p. 117-27.
[14] Cheng F, Zhang J, Wen C. Cost-sensitive large margin distribution machine for classification of imbalanced data. Pattern Recogn Lett. 2016;80:107-12.
[15] Wang S, Liu W, Wu J, Cao L. Training deep neural networks on imbalanced data sets. 2016 International joint conference on neural networks (IJCNN); 2016 Jul 24-29; Vancouver, Canada. New York: IEEE; 2016. p. 4368-74.
[16] Al-Azani SH, El-Alfy EM. Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short Arabic text. Procedia Comput Sci. 2017;109:359-66.
[17] Liu Y, Loh HT, Sun A. Imbalanced text classification: a term weighting approach. Expert Syst Appl. 2009; 36:690-701.
[18] Nguyen TT, Chang K, Hui SC. Supervised term weighting for sentiment analysis. Proceedings of 2011 IEEE international conference on intelligence and security informatics; 2011 Jul 10-12; Beijing, China. New York: IEEE; 2011. p. 89-94.
[19] Naderalvojoud B, Bozkir AS, Sezer EA. Investigation of term weighting schemes in classification of imbalanced texts. 8th European Conference Data Mining; 2014 Jul 15-17; Lisbon, Portugal. p. 39-46.
[20] Domeniconi G, Moro G, Pasolini R, Sartori C. A study on term weighting for text categorization: a novel supervised variant of tf.idf. Proceedings of 4th international conference on data management technologies and applications; 2015 Jul 20-22; Colmar, France. Portugal: SciTePress; 2015. p. 26-37.
[21] Triguero I, Rio S, Lopez V, Bacardit J, Benitez JM, Herrera F. ROSEFW-RF: the winner algorithm for the ECBDL'14 big data competition: an extremely imbalanced big data bioinformatics problem. Knowl Base Syst. 2015;87:69-79.
[22] Bird S, Klein EH, Loper E. Natural language processing with python. Sebastopol: O'Reilly Media; 2009.
[23] Li Y, Guo H, Zhang Q, Gu M, Yang J. Imbalanced text sentiment classification using universal and domain-specific knowledge. Knowl Base Syst. 2018;160:1-15.
[24] Mikolov T, Sukskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. Proceedings of the 27th international conference on neural information processing systems; 2013 Dec 5-19; Nevada, USA. New York: Curran Associates Inc; 2013. p. 3111-9.
[25] Ma L, Zhang Y. Using Word2Vec to process big text data. IEEE International Conference Big Data; 2015 Oct 29-Nov 1; Santa Clara, USA. New York: IEEE; 2015. p. 2895-7.
[26] Li J, Huang G, Fan C, Sun Z, Zhu H. Keyword extraction for short text via word2vec, doc2vec, and textrank. Turk J Electr Eng Comput Sci. 2019;27:1794-805.
[27] Dai K. PageRank Lecture Note [Internet]. 2009 [cited 15 Aug 2020]. Available from: https://www.ccs.neu.edu/home/daikeshi/notes/PageRank.pdf.
[28] Sibunruang C, Polpinij J. Concept-based text classification of Thai medicine recipes represented with ancient Isan language. In: Unger H, Meesad P, Boonkrong S, editors. Recent advances in information and communication technology. Berlin: Springer; 2015. p. 117-26.
[29] Lan M, Tan C, Su J, Lu Y. Supervised and traditional term weighting methods for automatic text categorization. IEEE Trans Pattern Anal Mach Intell. 2009;31(4):721-35.
[30] Chen K, Zhang Z, Long J, Zhang H. Turning from TF-IDF to TF-IGM for term weighting in text classification. Expert Syst Appl. 2016;66:1339-51.
[31] Carvalho F, Guedes GP. TF-IDFC-RF: a novel supervised term weighting scheme. arXiv:2003.07193. 2020:1-28.
[32] Atdag S, Labatut V. A comparison of named entity recognition tools applied to biographical texts. 2nd International conference on systems and computer science; 2015 Aug 26-27; Villeneuve d'Ascq, France. New York: IEEE; 2013. p. 228-33.
[33] Deng ZH, Luo KH, Yu HL. A study of supervised term weighting scheme for sentiment analysis. Expert Syst Appl. 2014;41:3506-13.
[34] Figueiredo F, Rocha L, Couto T, Salles T, Gonçalves MD, Wagner M. Word co-occurrence features for text classification. Inform Syst. 2011;36(5):843-53.
[35] Noumir Z, Honeine P, Richard C. On sample one-class classification methods. Proceedings IEEE International Symposium on Information Theory; 2012 Jul 1-6; Cambridge, USA. New York: IEEE; 2012. p. 2022-6.
[36] Bounsiar A, Madden MG. Kernels for one-class support vector machines. 2014 International Conference on Information Science & Applications (ICISA); 2014 May 6-9; Seoul, Korea. New York: IEEE; 2014. p. 1-4.