Current Applied Science and Technology Vol. 23 No. 2 (March-April 2023)

Research article

Automatically Correcting Noisy Labels for Improving Quality of Training Set in Domain-specific Sentiment Classification

Thananchai Khamket and Jantima Polpinij*

Intellect Laboratory, Faculty of Informatics, Mahasarakham University, Mahasarakham, Thailand

Received: 25 December 2021, Revised: 12 June 2022, Accepted: 10 August 2022

DOI: 10.55003/cast.2022.02.23.006

Abstract

Keywords

label noise;

sentiment classification;

polarity label analyzer;

k-NN;

logistic regression;

XGBoost;

linear SVM;

CNN

Classification model performance can be degraded by label noise in the training set. The sentiment classification domain also struggles with this issue, whereby customer reviews can be mislabeled. Some customers give a rating score for a product or service that is inconsistent with the review content. If business owners are only interested in the overall rating picture that includes mislabeling, this can lead to erroneous business decisions. Therefore, this issue became the main challenge of this study. If we assume that customer reviews with noisy labels in the training data are validated and corrected before the learning process, then the training set can generate a predictive model that returns a better result for the sentiment analysis or classification process. Therefore, we proposed a mechanism, called polarity label analyzer, to improve the quality of a training set with noisy labels before the learning process. The proposed polarity label analyzer was used to assign the polarity class of each sentence in a customer review, and then polarity class of that customer review was concluded by voting. In our experiment, datasets were downloaded from TripAdvisor and two linguistic experts helped to assign the correct labels of customer reviews as the ground truth. Sentiment classifiers were developed using the k-NN, Logistic Regression, XGBoost, Linear SVM and CNN algorithms. After comparing the results of the sentiment classifiers without training set improvement and the results with training set improvement, our proposed method improved the average scores of F1 and accuracy by 20.59%.

*Corresponding author: Tel.: (+66) 43654359 ext. 5365, 5003
E-mail: jantima.p@msu.ac.th

1. Introduction

Nowadays, people have easy access to online electronic media. Using the Internet to surf social media, search for information, check e-mails, watch television, listen to music or for online shopping are normal daily activities [1, 2]. Consequently, many businesses have turned to e-commerce to present data related to their products or services over the Internet to a global audience. Electronic buying or selling of products and services also has an immediate impact on revenue [3].

Effective and efficient e-commerce systems require recognition and comprehension of customer feedback to further fine-tune business opportunities. Accurate measurement of customer satisfaction can be used to develop the best customer experience, improve customer retention, provide satisfactory information for other consumers and optimize business decision-making [4, 5]. Therefore, many e-commerce systems provide a channel for customers to express their feelings (or opinions) as feedback about products and services.

Recognizing consumer needs and values can lead to business advantages. Recognizing customer needs or feedback can be done by the process of sentiment analysis (or opinion mining), which is the process of detecting positive or negative sentiment from text messages on social media sites to help business people understand consumer sentiment of their brand, product or service [6, 7]. This study field is an ongoing field of research in the text mining field, and it has been extensively studied and applied in a range of business and related domains [6, 8].

Issues involving sentiment analysis include data sparsity, multilingual aspects, emotion detection, subject detection and sarcasm detection [8]. However, most previous studies concerning sentiment analysis concentrated on developing more accurate sentiment classifiers to predict the sentiment polarity of segmentation [6, 9, 10-12]. When analyzing the classification of predictive modeling problems, training sets are very important for developing effective model data. A training set consists of examples collected from the problem domain, including input observations and output class labels. In sentiment classification (e.g. customer review classification), data collected from the problem domain can be mislabeled [13], with customers giving a rating score for a product or service that is inconsistent with the review content. If business owners are only interested in the overall rating picture that includes mislabeling, this can lead to misunderstandings and erroneous business decisions.

In the domain of automatic sentiment analysis, if the training set contains mislabeling or noisy label, this may also lead to ineffective sentiment classifiers that return poor results of opinion polarity predictions [14, 15]. Currently, three possible solutions can be applied to handle this issue. First, data having label noise or no label are removed [16]. Second, a method is presented for handling label noise during learning the predictive model [17, 18]. This solution is to apply an algorithm or method to re-label data before learning predictive model. For an example, Tomek [17] applied the *k*-nearest neighbor for re-labelling data. Lastly, labels of noisy training data are automatically validated and corrected before learning the predictive model. Unfortunately, different noisy label datasets may require different methods for solving an issue [19]. A solution that is suitable for some datasets might not be suitable for other specific datasets, including textual sentiment datasets. Consequently, this challenge is addressed here. We consider that if customer reviews with noisy (or mislabeled) labels in training data are validated and corrected before the learning process, that training set might be used to generate a predictive model that can return a better result for the sentiment analysis or classification process.

The performance of classification models can be degraded by label noise, while the fundamental reliability of supervised learning also depends on training labels [13, 16-23]. This issue was first addressed in 1972 [16]. The first solution that was proposed for handling this issue was to remove noisy data labels from the training set before developing classification models, and the experiments illustrated that filtering significantly improved classification accuracy for noise levels

up to 30% [13]. Later, many researchers mentioned the danger of automatically removing instances. They commented that the instances might not be correctly classified, with some exceptions to the general rule appearing to be incorrectly labeled [16, 20]. A key question is how to improve data quality in a training set having noisy labels [16, 17].

Guyon *et al*. [18] utilized information criterion to measure sample data by presenting irregular instances to a domain expert to identify whether they were mislabeled or exceptions. However, they remarked that the ordering process might affect their online method. Oka and Yoshida [19] proposed a method for separating generalizations and exceptions by maintaining a record of data inputs that were correctly and incorrectly classified. The main mechanism used for identifying noisy labels from exceptions was driven on a user-specified parameter to guarantee that the classification rate of each stored sample was satisfactory. However, their approach only involved experiments on artificial datasets. Natarajan *et al*. [20] proposed the methods of unbiased estimators and weighted loss functions to solve the issue of risk minimization in the presence of random classification noise. They also developed efficient algorithms for those methods, with proven guarantees for the learning of classification models under label noise. They stated that their proposed algorithms were easy to implement and helped to impressively improve the classification performance, even at high noise rates.

Liu and Tao [21] addressed the problem of label noise by re-weighting frameworks for classification. Theoretical analyses were performed to make sure that the learned classifiers were optimal for the noise-free sample. After experimentation with the proposed learning framework for synthetic and real-world datasets, the results showed that this framework was effective and robust. Furthermore, this work preposed a method for estimating the noise rates.

Moreover, it has been confirmed that noise issues severely degrade the general performance of deep learning algorithms in classification domains [22-27]. Therefore, the issue of noisy labels continues to be studied using deep machine learning. In 2015, Reed *et al*. [22] proposed a method for handling noisy labels by augmenting prediction objectives with a notion of consistency. That is, if the same prediction gave similar percepts, where the notion of similarity was between deep network features computed from the input data, then the prediction was consistent. After experimenting on several datasets, the results showed that this approach gave substantial robustness to label noise, especially on MNIST handwritten digits. Furthermore, their model might have been robust for labelling corruption.

In 2019, Han *et al*. [23] proposed an iterative self-learning framework for real noisy datasets. They proved that a single prototype was inadequate to represent a class's distribution and multi-prototypes were essential. They also backed up their claim that original noisy labels were helpful in the learning process, although corrected labels were more precise. By correcting the label using several class prototypes and iteratively training the network using the corrected and original noise, their proposal provided an effective end-to-end training process without using an accessorial network or adding extra supervision on a real noisy dataset.

The issue of noisy labels also effects all study areas (i.e. sentiment classification) in the domain of natural language processing (NLP) because it has been found that this field also suffers from noisy labels due to erroneous automatic and human annotation procedures [28-31]. The study of classifiction with noisy labels is still of interest because a solution that is suitable for some datasets may not be suitable for other specific datasets, e.g., the domain-specific sentiment classification. To the best of our knowledge, there are few studies related to the issue of noisy labels in the domain-specific sentiment classification. Examples of relevant studies are as follows.

Malik and Bhardwaj [30] proposed a method of manual label correction. Their results showed that the validation and correction of labels by domain experts helped to enhance the Micro and Macro-F1 scores acquired by Linear SVMs by as much as 14.5% and 30%, respectively. Then, they concentrated on a collection of professionally labeled news stories. However, hand-crafted label correction is time-consuming process and costly. Eamwiwat *et al*. [31] proposed three

contributions to the implementation of social analysis models. First, text pre-processing could be used to relieve noise or outlier from input textual data. Second, robustness towards word segmentation was improved by using an ensemble method with two tokenizers. Third, a training process inspired by the co-training method was proposed in order to filter label noise within the data, and then those training documents were re-labelled through probabilistic prediction from a trained model. After that, the re-labelled documents were used to train a new model. Their model improved the average Macro-F1 score by 2.56% when compared with the baseline models peformed on social media data.

From the above discussion, it can be seen that there have been a number of solutions proposed for addressing the issue. This is because different approaches may be required to address the issue of noisy labels in different data domains. Simply speaking, a solution that is suitable for some datasets may not be suitable for other specific datasets, including textual sentiment datasets. We hypothesized that earlier studies focused only on correcting class labels prior to or during predictive model building. Those studies might not have considered solutions under different ratios of incorrect class labels in thier datasets. If those proposed methods have been applied to datasets with fewer noisy labels, they might have returned better results. Furthermore, those proposed methods might have returned worse results if they had been applied to data sets containing a lot of noisy labels. Based on this assumption, we proposed a study that experiments with various ratios of noisy labels in our dataset.

2. Materials and Methods

2.1 Dataset

The three datasets used in this study, which were customer reviews relating to hotels, were gathered from the TripAdvisor website between December 2020 and May 2021. Each customer review was based on a 5-star rating scale. Our datasets were stored in CSV format.

The first dataset was used to model the predictive classifier for correcting labels of noisy training data before applying the learning sentiment classifier model. A total of 500 customer reviews with rating scores of 1 and 2 were assigned to the negative class, while another 500 customer reviews with rating scores of 4 and 5 were assigned to the positive class. Two linguistic experts validated the correctness of the polarity class of each customer review to guarantee no customer reviews with noisy labels. This dataset was used for developing the predictive model, called a polarity label analyzer, which was used to validate and correct (re-label) the polarity class of custom reviews.

The second dataset was used for the experimental process. In this dataset, 200 customer reviews with the correct polarity label per class and 200 customer reviews with incorrect polarity label per class were provided. It was noted that customer reviews with incorrect polarity label were also given the correct polarity class by the domain expert as well, where they could be used as the ground truth. To generate a training set, we randomly selected customer reviews from each class using ratios of customer reviews with correct polarity and customer reviews with incorrect polarity for a binary-class classification as 10:5, 10:4, 10:3, 10:2 and 10:1, respectively. This dataset was used in the experimental stage.

The third dataset was used as the test set. Similar to the first dataset, 200 customer reviews with rating scores 1 and 2 were assigned to the negative class, while 200 customer reviews with rating scores 4 and 5 were assigned to the positive class. These customer reviews had noisy labels because the contents were inconsistent with the original rating scores. Therefore, when assigning them to positive or negative classes based on their rating scores, the reviews could be assigned to

the wrong polarity class. Thus, two linguistic experts also helped to validate and assign the correct labels for the customer reviews in this dataset as the ground truth. It was noted that only the customer reviews for which the two experts had assigned the same result were chosen. Some examples are shown in Table 1. This dataset was used to test and consider the performance of the noisy label correction of the predictive model generated from the first dataset. Also, the third dataset was used to test and consider the performance of the predictive models generated from the second dataset. This involved a comparisons of the predictive models generated from the second dataset without noise label improvement in the training set and the predictive models generated from the second dataset with noise label improvement in the training set.

Table 1. Examples of customer reviews corrected polarity label

ID	Examples of customer reviews	Original Polarity Label	Corrected Polarity Label
1.	The bathroom in the hotel is quite clean and the towel like sandpaper. What is a nice service!!!	Positive	Negative
2.	This hotel is an exciting place I was awfully thrilled all time when staying there.	Positive	Negative
3.	The staff is friendly. Room is small but clean. The location is in the middle of the city. It is better if the parking is large.	Negative	Positive

2.2 Preliminary: Developing the polarity label analyzer

This section describes a method for developing a predictive model to improve the quality of training sets by correcting polarity labels of customer reviews in the training sets before the application of learning classification models. The predictive model for correcting polarity labels was developed using sentence-level sentiment analysis. By analyzing the polarity of each sentence in a document and then concluding the polarity class of that customer review by voting, this may help to assign a correct and appropriate polarity sentiment to a considered customer review. This involves sentences that express a single opinion to define their orientation [32, 33]. On the other hand, the document-level sentiment analysis just determines the overall opinion of the document. This could result in an erroneous reflection of the customer review polarity sentiment.

This stage used the first dataset to develop the predictive model, called "*polarity label analyzer*". We utilized the Natural Language Processing Toolkit (aka NLTK) which is a Python package for developing the polarity label analyzer. Each processing step in the proposed method for developing the polarity label analyzer can be described as follows and the overview of the method of developing the polarity label analyzer is shown in Figure 1.

2.2.1 Pre-processing of customer reviews

Pre-processing involves converting raw data into a representation suitable for application. This process involves several steps: tokenization, text cleaning by removing special characters, conversion of all characters into lower case (i.e. "Happy" to "happy"), expansion of contractions (i.e. "isn't" to "is not"), expansion of abbreviations, stemming by the snowball technique, stop word removal, and finally feature selection. The training set was then represented in the format of a vector space model (VSM). In this study, features in the context of sentiment analysis were words used for expressing opinions, either positive or negative.

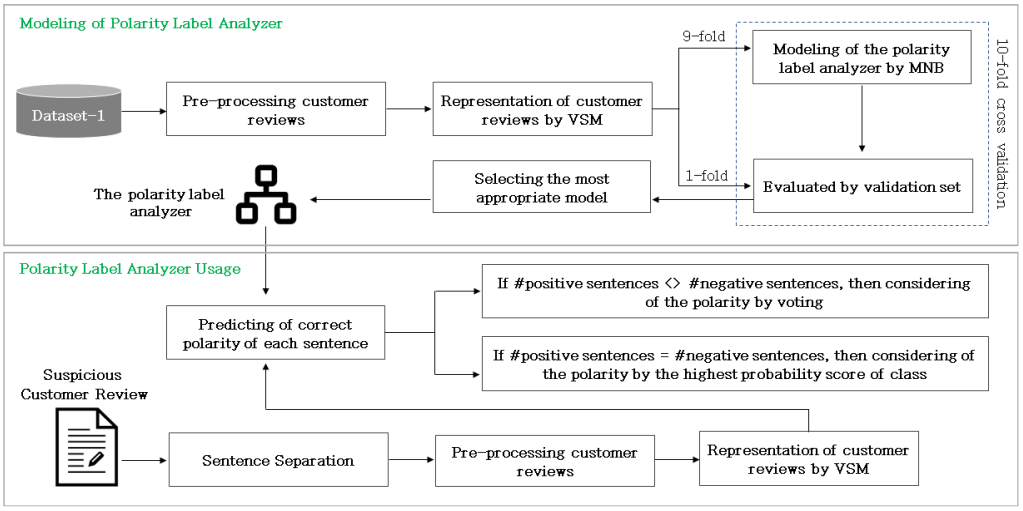

Figure 1. Method of developing the polarity label analyzer

To obtain the most appropriate words for predictive modeling, we applied the filter method to select features based on *information gain* (*IG*) [34, 35]. If words having IG scores are greater than or equal to 0.05 [34], these words are chosen as features. Each selected word (or feature) is then weighted by the *term frequency-inverse document frequency* (*tf-idf*) scheme. The *term frequency (tf)* can be defined *as tf*(w_i, d_j), where it is the number of times that word w_i appears in a customer review document d_j. To normalize the *tf*, the equation of *tf* should be:

$$tf(w_i, d_j) = \log(1 + tf(w_i, d_j)) \tag{1}$$

Meanwhile, *idf*(w_i) is a frequently appearing word w_i in a customer review collection. The equation of *idf*(w_i) can be defined as:

$$idf(w_i) = \log(1 + \frac{|D|}{df(w_i)}) \tag{2}$$

In this study, $|D|$ is the total number of customer reviews in the entire customer review collection, while *df*(w_i) is the number of customer reviews in the collection containing word w_i. In fact, *idf*(w_i) is generally used to evaluate the importance of a word to the collection by giving a reliable weight score to the rare word [35].

2.2.2 Modeling of polarity label analyzer

We applied a well-known supervised machine learning algorithm, the Multinomial Naïve Bayes (MNB), to develop the polarity label analyzer used for predictive models. This algorithm is a popular method of text classification.

The MNB is a probabilistic algorithm that is mostly applied in text classification task [35]. This algorithm is based on the Bayes theorem. It estimates the probability of each class for a given sample, and then returns the class with a probability score as output. The pseudocode of MNB used in this study can be presented as Figure 2.

Function Train MultinomailNaïveBayes(D, C)
Return $P(c)\times P(w|c)$
for each class $c \in C$ #Estimate $P(c)$
N_{doc} = Total of customer reviews in D
N_c = Total of customer reviews from D in class c

$$prior[c] \leftarrow \frac{N_c}{N_{doc}}$$

$V \leftarrow$ vocabulary of D
$doc[c] \leftarrow$ **append**(d) **for** $d \in D$ **with** class c
for each term-word w in V #Estimate $P(w|c)$
$count(w, c) \leftarrow$ #of occurrences of w in $doc[c]$

$$likelihood[w, c] \leftarrow \frac{count(w,c)+1}{\sum_{w' in V}(count(w',c)+1)}$$

return *prior*, *likelihood, V*

Figure 2. The pseudocode of multinomial naïve bayes

2.2.3 Utilizing the polarity label analyzer

The polarity label analyzer was used to correct the polarity class of customer reviews by recognizing the polarity of a sentence (e.g. positive and negative) as follows. Given a set of sentences of a customer review document D, each sentence S contained a set of words W.

Next, the polarity label analyzer analyzed the polarity class of each sentence S, and the concluded polarity class of that customer review was obtained by voting. Simply speaking, if a customer review had the number of positive sentences appearances greater than the number of negative sentences appearances, then this review was assigned to the positive class. By contrast, if a customer review had the number of negative sentence appearances greater than the number of positive sentences appearances, then this review was assigned to the negative class.

However, if the number of positive sentence appearances was equal to the number of negative sentences appearances, the probability score was utilized to determine the sum of the probability scores for each sentence in each class. The predicted polarity class label for a customer review was considered as the polarity class label with the highest probability score.

2.3 Experimental setup

In this stage, the third dataset was used for the experiment. Customer reviews with incorrect polarity labels were given the correct polarity class by the domain experts as the ground truth. The performances of predictive models developed using training sets that were not corrected for polarity class labels and predictive models developed using training sets that were corrected for polarity class labels were compared.

2.3.1 Pre-processing of customer reviews

There were two different stages of pre-processing for customer reviews (Figure 3). The first stage involved the pre-processing of the training set without correcting the polarity label. The second was about pre-processing the training set by correcting the polarity label based on the use of the polarity label analyzer.

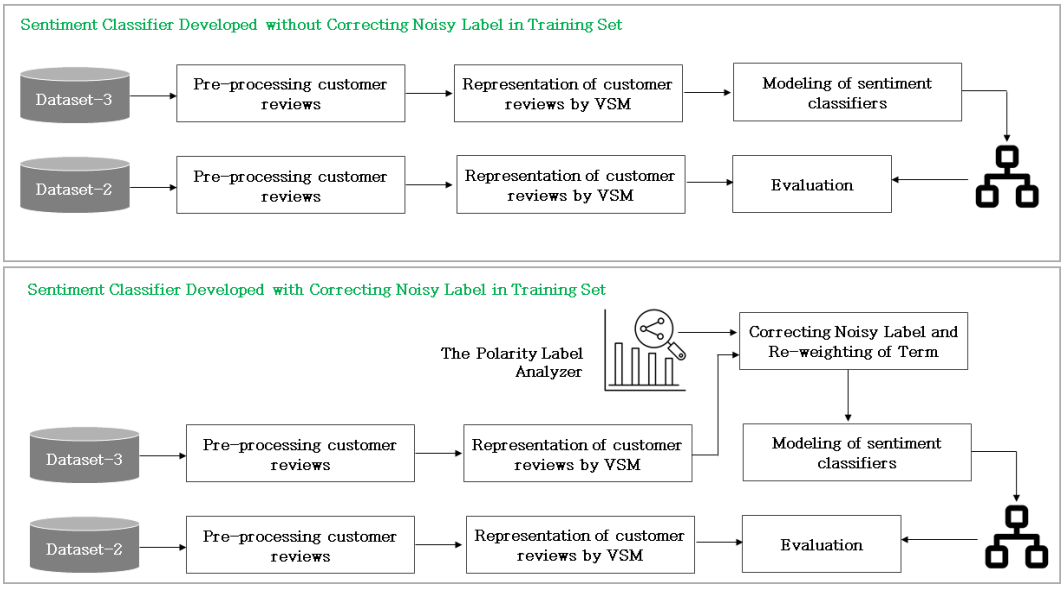

Figure 3. An overview of experimental setup

1) Pre-processing the training set without correcting the polarity label

In general, the training set was first performed following the pre-processing process, commencing with tokenization, and then text cleaning to remove special characters, convert all letters to lower case, expand contractions and abbreviations, stem by the snowball technique, remove stop words, and lastly select features by IG. Finally, the training set was represented in VSM format, with each word (or feature) weighted by the *tf-idf* scheme. The process of modeling sentiment classifiers was then performed.

2) Pre-processing the training set with correcting the polarity label

Pre-processing the training set followed a similar method to pre-processing predictive models from the training set without correcting the polarity class labels. However, before applying the learning sentiment classifier models, the training set was corrected for polarity class labels by the polarity label analyzer, in a process called "*correcting noisy label*". After obtaining the improved training set with the corrected polarity class label, the improved training set was then given the new weight of each word using the *tf-idf* scheme, called "*re-weigting of term*". This is because when documents are given a new class label, the documents in each class may also change accordingly. Therefore, the "words" used as document features of each class are subject to change. Furthermore, the weighting of the features needs to be updated to determine the actual weight of each feature that truly corresponds to the documents in each class. Afterwards, the process of modeling sentiment classifiers was then performed.

2.3.2 Modeling of sentiment classifiers

For modeling of sentiment classifiers, we applied four supervised machine learning algorithms: *k*-Nearest Neighbors (*k*-NN), Logistic Regression (LR), eXtreme Gradient Boosting (XGBoost) and

SVM. Furthermore, a Convolutional Neural Network (CNN) which is a deep learning algorithm was also used in our experiment.

1) *k*-Nearest neighbors (*k*-NN)

k-NN is a non-parametric algorithm used for classification tasks that assumes similar objects exist in close proximity [35, 36]. Accordingly, *k*-NN positions and ranks the nearest *k* neighbors of the labeled examples from the training dataset and utilizes the classes of the highest-ranked neighbors to consider a class assignment. The nearer that neighbors are within the same class, the higher the confidence in that prediction. This study takes the case of $k = 3$.

2) Logistic regression (LR)

The LR algorithm is applied to classify individuals in categories according to logistic function. There are many instances where a perfect graph that fits all the data points is not apparent [37]. A learn textual sentiment classifier, denoted as $y = f(x)$ can be obtained from a training set, denoted as $D = \{(x_1, y_1), \ldots, (x_i, y_i)\}$. For textual sentiment classification, the vectors $x_i = [x_{i1}, \ldots, x_{ij}, \ldots x_{id}]^T$ comprise transformed word weights from customer reviews. The values $y_i = \in \{-1,+1\}$ are class labels encoding positive polarity (+1) or negative polarity (-1) of the vector in the class. Class labels as -1/+1 rather than 0/1 were encoded to simplify the presentation of our fitting algorithm. The LR algorithm was denoted as a conditional probability model with the following form:

$$p(y_i = +1 \mid \beta, x_i) = \frac{1}{1 + \exp(-\beta^T x_i)} \tag{3}$$

For a textual sentiment classification task, $p(y = +1|x_i)$ corresponds to the probability that the *i*-th customer review belongs to the class label. Class label assignment was based on comparing the probability estimate with a threshold. Also, class label assignment could be more generally based on maximizing the expected effectiveness.

3) Support vector machines (SVM)

SVM finds the optimal linear separator between data points with a maximum margin that allows positive values greater than the margin and negative values less than the margin [37, 38]. This method is called quadratic programming optimization. Let the training set be denoted as $\{(x_{11}, y_1), (x_{12}, y_2), \ldots, (x_{mn}, y_m)\}$, where x_{ij} is the occurrence of event *j* in time *i*, and $y_i \in \{-1, 1\}$. The SVM algorithm solves the following quadratic problem:

$$\min_{\xi, w, b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i \tag{4}$$

Subject to:

$$y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad \xi_i \geq 0; i = 1, \ldots, m$$

where ξ_i is the slack variable in which there are non-separable cases, while $C > 0$ if the difference between the margin and the sum of errors is controlled by the soft margin. Simply speaking, it imposes a penalty on data that has been misclassified (misclassification). The penalty increases as

the distance to the margin lengthens, while *w* is the hyperplane's slope that is used to separate the data [38].

The strength of SVM comes from its ability to apply a linear separation on high dimension non-linear input data, obtained by employing a suitable kernel function [38]. This study used linear kernels for our textual sentiment classification because most of text classification problems are linearly separable.

4) eXtreme gradient boosting (XGBoost)

XGBoost [37] is an ensemble learning algorithm like Random Forest. It is a type of gradient-boosted decision tree classifier that can predict any kind of data from previously predicted data. In XGBoost, the trees are built sequentially and the the errors of the previous tree can be reduced in each subsequent tree. These subsequent trees are called weak learners. Each of these weak learners contributes some crucial information for prediction, allowing the boosting concept to build a strong learner by effectively combining the weak learners. The strength of XGBoost is its scalability, which enables rapid learning via parallel and distributed computation while still ensuring optimal memory utilization. This study generated 100 decision trees for our predictive model.

5) Convolutional neural network (CNN)

CNN [39] comprises one input layer, multiple hidden layers and one output layer. Connections between nodes are not cyclical. A CNN uses a variety of multilayer perceptrons that require minimal pre-processing. CNNs are most commonly applied for image analytics but have also proved useful for text classification. The CNN architecture for text classification generally consists of four connected layers. They are a word embedding layer, a convolutional layer, a max-pooling layer, and a softmax layer. The setting of CNN in this study is described in Figure 4. It should be noted that when devolving a sentiment classifier model using a CNN, a *tf-idf* weighting scheme is not required for CNN text classifier modeling.

Word embedding as the first layer in CNN is used to transform text into a meaningful numerical form based on vector representation. This layer maps words to 1-V representation, where V is vocabulary size, and then uses hidden layers to learn semantic representation from word vectors. The hidden layers of a neural network essentially act as a feature extractor, transforming word vectors into lower dimensional vector space and encoding the semantics of the words.

A convolutional layer is used to convert the texts to sequences of word embeddings as input. This layer uses "*convolution filters*" to create feature vectors by analyzing the word embeddings for each text. Simply speaking, the convolutional layer analyzes the word embeddings to find convolutions and further reduce dimensional complexity and computation by the max-pooling layer.

The max-pooling layer utilizes the variable-length feature vectors obtained from the convolutional layer as input and produces fixed-length vectors. Consequnetly, the less-relevant local information should be ignored and removed.

The softmax layer is utilized to transform fixed-length feature vectors to be the input to the fully-connected layer. The output of this fully-connected layer is the value of each class. The softmax layer uses the softmax activation function for forcing the output of the CNN to indicate predicted probabilities for each of the classes. The class that achieves the highest prediction probability is the result prediction class that is generated from CNN.

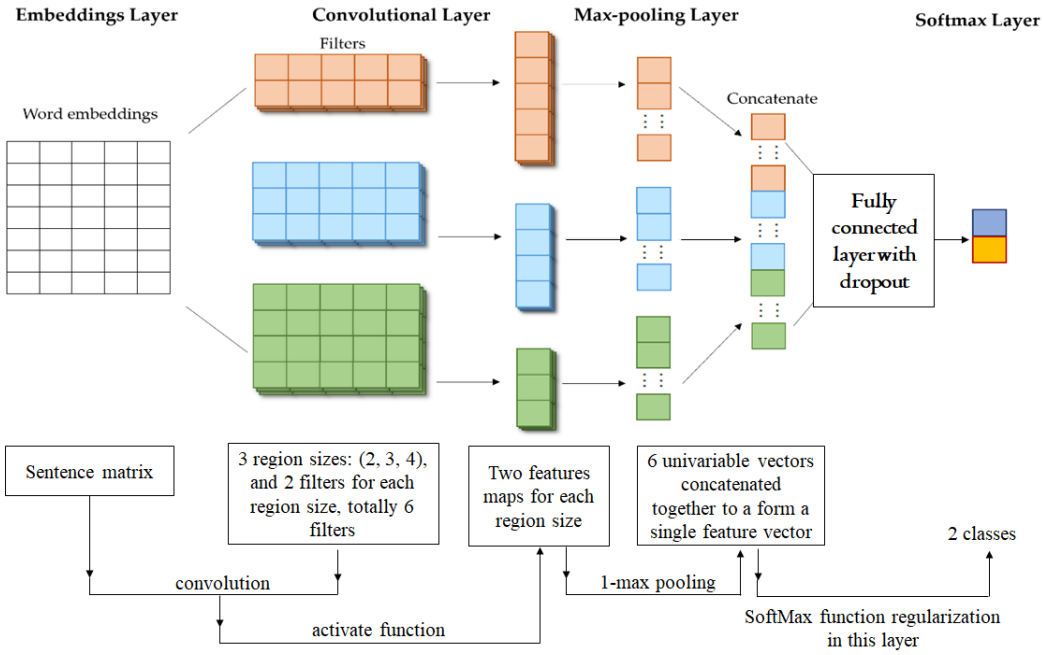

Figure 4. The architecture of a CNN for text classification

3. Results and Discussion

This study details two evaluation stages. First, label noise correction performance was assessed to select the best predictive models for use in the next stage. Second, the sentiment classification performance of predictive models developed using a training set that was not corrected for polarity class label and predictive models developed using a training set that was corrected for polarity class label were evaluated and compared using the polarity label analyzer.

3.1 Evaluation of the polarity label analyzer

Two measurement techniques were applied to evaluate the performance of the polarity label analyzer as F1 and accuracy [38, 39]. The highest scores of F1 and accuracy were selected as optimal for correcting polarity label noise in the next stage. We set up our experiments with 10-fold cross-validation. A single subsample was retained as the validation set for testing the model, while the remaining 9 subsamples were used as training data. To increase the confidence of noisy label correction, the selected model was also tested by the second dataset. The experimental results are shown in Table 2. The results show that the polarity label analyzer model built in round 5 gave the optimal performance for correcting label noise of customer reviews compared to the ground truths established by the domain experts. However, the main factor behind errors was identified as being sarcastic customer reviews that were very hard to automatically analyze for their correct label. Therefore, a semantic function should be developed for label noise correction in future studies.

The best predictive model as revealed by the polarity label analyzer was chosen and used for correcting customer reviews with noisy labels in the next stage.

Table 2. The experimental results of the polarity label analyzer

10-fold cross validation	Validation Set		Test Set	
	F1	Accuracy	F1	Accuracy
Round 1	0.82	0.80	0.79	0.78
Round 2	0.82	0.80	0.80	0.79
Round 3	0.67	0.67	0.65	0.65
Round 4	0.71	0.73	0.70	0.71
Round 5	**0.93**	**0.93**	**0.91**	**0.91**
Round 6	0.82	0.80	0.79	0.78
Round 7	0.88	0.87	0.86	0.85
Round 8	0.88	0.87	0.86	0.85
Round 9	0.86	0.86	0.86	0.86
Round 10	0.75	0.75	0.74	0.74

3.2 Evaluation of sentiment classifier models

In this stage, we evaluated and compared the performance of sentiment classification between predictive models developed using a training set that had not been corrected for polarity class labels and predictive models developed using a training set that was corrected for polarity class label using the polarity label analyzer. F1 and accuracy were also used for the evaluation. The results are shown in Table 3. The results show that sentiment classifiers built from the training set with improved quality return gave better results than sentiment classifiers built from the training set without improved quality. This was because the proposed method can improve *true positive* (*tp*) and *true negative* (*tn*) scores. Therefore, it helps to improve the accuracy and F1 scores accordingly. The average score of F1 and accuracy of the models built by the training set without correction of label noise was 0.68, while the average score of F1 and accuracy of the models built by the training set with correction of label noise was 0.82. It can be seen that the average scores of accuracy and F1 improved by 20.59%. The highest accuracy and F1 scores were achieved with linear SVM, while *k*-NN had the lowest performance. The CNN and XGBoost algorithms also returned satisfactory results but they required more time than the other algorithms for training and validating the classifier models. In the case of the XGBoost algorithm, the algorithm was based on a bagging algorithm using ensemble learning concepts. The large number of trees made this algorithm slow and inefficient for computational prediction. Meanwhile, the slow computational processioning of the CNN was due to the underlying complex mathematical operations. Furthermore, the CNN returned poorer results than linear SVM because all deep learning algorithms generally requires a lot of training data. Unfortunately, the dataset used as a training set in this study was small; therefore, it was impossible to train the effective sentiment classifiers using the CNN with the small dataset.

Some customer reviews were sarcastic and these were not easy to automatically analyze and assign correct polarity labels. This is an inherent ambiguity problem in natural language as an intrinsic characteristic of human conversations, and is particularly challenging in natural language understanding (NLU) scenarios. A semantic function will require further analysis of label noise correction in future studies. However, one solution to address the problem of sarcastic customer reviews is to utilize emoticons in customer reviews. Many customer reviews include these special characters. These emotions should not be removed but used as extra data in the analysis process. In addition, using a supervised term weighting scheme, i.e. *term frequency-inverse gravity moment* (*tf-igm*), instead of *tf-igm*, may help to improve the performance of noise label correction and sentiment classifiers because the supervised term weighting scheme can precisely calculate the distinguishing class of a term. By contrast, *tf-idf* is an unsupervised term weighting scheme that may struggle to accurately specify the class discriminative power of each word.

Table 3. The results of comparing the performance of sentiment classifiers

Algorithms	Ratio of customer reviews with correct label and customer reviews with noisy label in training set	Sentiment classifiers built by training set without correcting label noise		Sentiment classifiers built by training set with correcting label noise	
		F1	Accuracy	F1	Accuracy
k-NN	10:1	0.72	0.72	0.79	0.79
	10:2	0.68	0.68	0.78	0.78
	10:3	0.65	0.65	0.77	0.77
	10:4	0.56	0.56	0.77	0.77
	10:5	0.52	0.52	0.77	0.77
LR	10:1	0.75	0.75	0.82	0.82
	10:2	0.71	0.71	0.82	0.82
	10:3	0.68	0.68	0.81	0.81
	10:4	0.61	0.61	0.80	0.80
	10:5	0.57	0.57	0.79	0.79
Linear SVM	10:1	**0.77**	**0.77**	**0.88**	**0.88**
	10:2	**0.75**	**0.75**	**0.88**	**0.88**
	10:3	**0.73**	**0.73**	**0.87**	**0.87**
	10:4	**0.67**	**0.67**	**0.86**	**0.86**
	10:5	**0.64**	**0.64**	**0.86**	**0.86**
XGBoost	10:1	0.76	0.76	0.85	0.85
	10:2	0.74	0.74	0.85	0.85
	10:3	0.71	0.71	0.84	0.84
	10:4	0.68	0.68	0.84	0.84
	10:5	0.62	0.62	0.83	0.83
CNN	10:1	0.76	0.76	0.85	0.85
	10:2	0.74	0.74	0.85	0.85
	10:3	0.71	0.71	0.85	0.85
	10:4	0.65	0.65	0.84	0.84
	10:5	0.65	0.65	0.84	0.84
	Average Scores	0.68	0.68	0.83	0.83

3.3 Comparing the proposed method and the baseline

Significantly, the best model for our proposed method was also compared with the baseline proposed by Wang *et al*. [29]. The baseline presented a method to deal with noisy labels during training, which was called a convolutional neural NETwork with AB-networks (NETAB). The NETAB consists of two convolutional neural networks (CNNs). They were the A-network and the AB-network. The A-network was used for learning sentiment scores to predict "*clean*" labels, while the AB-network was used for learning a noise transition matrix to handle input noisy labels. We also used the pre-trained embedding GloVe.840B downloaded from https://nlp.stanford.edu/projects/glove/. It was proposed by Pennington *et al*. [40] to initial the word vector, while the embedding dimension was 300. The batch size of the NETAB setting was kept constant at 50, the number of epochs was 100, the value of dropout was 0.5 and the window length was 5 with 100 features maps per window size. The input length of sentence is set to 40. We used 'Adam' as a default for the optimization function with a learning rate of 0.001. We also used our dataset for this comparison. The results of comparison are shown in Table 4.

Table 4. The results of comparing performances between the proposed method and the baseline

Methods	Ratio of customer reviews with correct label and customer reviews with noisy label in training set	Performance	
		F1	Accuracy
Proposed Method	10:1	0.88	0.88
	10:2	0.88	0.88
	10:3	0.87	0.87
	10:4	0.86	0.86
	10:5	0.86	0.86
NETAB	10:1	0.82	0.82
	10:2	0.82	0.81
	10:3	0.80	0.80
	10:4	0.79	0.79
	10:5	0.77	0.77

Table 4 showed that our method returned better results than the baseline. There were probably three plausible explanations. Firstly, the polarity label analyzer helped to improve the quality of the training set before modeling sentiment classifiers were applied. Secondly, linear SVMs could generalize well in high-dimensional feature spaces, making text classification easier to apply, with also the advantage of being more resilient than other approaches. Lastly, *tf-igm* used as term weighting scheme helped to increase the distinguishing power of term classes. In this way, the use of linear SVM together with *tf-igm* can improve the efficiency of data classification. Although our proposed method gave improved average scores of F1 and accuracy, it may not be superior to the baseline from every viewpoint. Our use of the small dataset might be a main reason that the baseline returned lower results than our proposed method. If we had utilized the same dataset or algorithm settings as the baseline method, an impact on the experimental results would likely have been observed.

4. Conclusions

This study proposed a method for improving sentiment classification with label noise in a training set. The main idea of our method was to validate and correct noisy data labels before the learning process took place. This improved the training used to generate a predictive model with a better result for sentiment classification. Three datasets downloaded from TripAdvisor were used in this study and two linguistic experts helped to give the correct polarity labels of the customer reviews to use as the ground truth.

First, we developed a mechanism that was called a polarity label analyzer and was based on MNB in order to validate and correct label noise in a training set before using the data to train the sentiment classifiers. To correct label noise, the polarity label analyzer was used to analyze the polarity class of each sentence in a customer review. The concluded polarity class of that customer review was obtained by voting. However, if the number of positive sentence appearances was equal to the number of negative sentences appearances, the probability score was utilized to determine the sum of the probability scores for each sentence in each class. The predicted polarity class label for a customer review was considered as the polarity class label with the highest probability score.

Finally, we used the polarity label analyzer to correct noisy labels before applying the learning process. Sentiment classification with improved noisy labels in the training set was compared with sentiment classification without improving the noisy labels in the training set. We developed sentiment classifiers using *k*-NN, LR, XGBoost, SVM and CNN algorithms. After

comparing the sentiment classifier results without correcting label noise in the training set and results of sentiment classifiers with corrected label noise in the training set, our proposed method improved average scores of F1 and by 20.59%. This shows that our proposed method could improve the quality of sentiment classification with label noise in a training set.

5. Acknowledgements

This research was financially supported by Mahasarakham University.

References

[1] Kaushik, R., 2012. Impact of social media on marketing. *International Journal of Computational Engineering and Management*, 15(2), 91-95.
[2] Appel, G., Grewal, L., Hadi, R. and Stephen, A.T., 2020. The future of social media in marketing. *Journal of the Academy of Marketing Science*, 48, 79-95.
[3] Chong, A.Y.L., Lacka, E., Li, B. and Chan, H.K., 2018. The role of social media in enhancing guanxi and perceived effectiveness of E-commerce institutional mechanisms in online marketplace. *Journal of Information and Management*, 55(5), 621-632.
[4] He, W., Wang, F.-K. and Akula, V., 2017. Managing extracted knowledge from big social media data for business decision making. *Journal of Knowledge Management*, 21(2), 275-294.
[5] Karakaya, F. and Barnes, N.G., 2017. Impact of online reviews of customer care experience on brand or company selection. *Journal of Consumer Marketing*, 27(5), 447-457.
[6] Medhat, W., Hassan, A. and Korashy, H., 2017. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), 1093-1113.
[7] Feldman, R., 2013. Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4), 82-89.
[8] Mohammad, S.M., 2021. *Sentiment Analysis: Automatically Detecting Valence, Emotions, and Other Affectual States from Text*. [online] Available at: https://arxiv.org/pdf/2005. 11882.pdf.
[9] Mohammad, S.M., 2017. Challenges in sentiment analysis. In: E. Cambria, D. Das, S. Bandyopadhyay and A. Feraco, eds. *A Practical Guide to Sentiment Analysis*. Cham: Springer, pp. 61-83.
[10] Pang, B., Lee, L. and Vaithyanathan, S., 2002. Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, Philadelphia, USA, July 6-7, 2002, pp. 79-86.
[11] Ye, X., Dai, H., Dong, L.-A. and Wang, X., 2021. Multi-view ensemble learning method for microblog sentiment classification. *Expert Systems with Applications*, 166, DOI : 10.1016/j. eswa.2020.113987.
[12] Alamoudi, E.S. and Alghamdi, N.S., 2021. Sentiment classification and aspect-based sentiment analysis on yelp reviews using deep learning and word embeddings. *Journal of Decision Systems*, 30(2-3), 259-281.
[13] Brodley, C.E. and Friedl, M.A., 1999. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11, 131-167.
[14] Wang, H., Liu, B., Li, C., Yang, Y. and Li, T., 2019. *Learning with Noisy Labels for Sentence-Level Sentiment Classification*. [online] Available at: https://arxiv.org/abs/1909.00124.
[15] Wang, L., Xu, X., Guo, K. and Cai, B., 2018. Visual sentiment analysis with noisy labels by reweighting loss. *2018 IEEE International Conference on Systems, Man, and Cybernetics* (*SMC*), Miyazaki, Japan, October 7-10, 2018, pp. 1873-1878.

[16] Wilson, D., 1972. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2(3), 408-421.
[17] Tomek, I., 1976. An experiment with edited nearest-neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics*, 6(6), 448-452.
[18] Guyon, I., Matic, N. and Vapnik, V., 1996. Discovering informative patterns and data cleaning. In: U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurasamy, eds. *Advances in Knowledge Discovery and Data Mining*. Cambridge: AAAI/MIT Press, pp. 181-203.
[19] Oka, N. and Yoshida, K., 1996. A noise-tolerant hybrid model of a global and a local learning model. *Proceedings of the AAAI-96 Workshop: Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, Oregon, USA, August 2-8, 1966, pp. 95-100.
[20] Natarajan, N., Dhillon, I.S., Ravikumar, P.K., and Tewari, A., 2013. Learning with noisy labels. In: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, eds. *Advances in Neural Information Processing Systems*. New York: Curran Associates, Inc., pp. 1196-1204.
[21] Liu, T. and Tao, D., 2016. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3), 447-461.
[22] Reed, S.E., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A., 2015. *Training Deep Neural Networks on Noisy Labels with Bootstrapping*. [online] Available at: https://arxiv.org/abs/1412.6596.
[23] Han, J., Luo, P. and Wang, X., 2019. *Deep Self-learning from Noisy Labels*. [online] Available at: https://arxiv.org/abs/1908.02160.
[24] Sanchez, G., Guis, V., Marxer, R. and Bouchara, F., 2020. *Deep Learning Classification with Noisy Labels*. [online] Available at: https://arxiv.org/abs/2004.11116.
[25] Cordeiro, F.R. and Carneiro, G., 2020. *A Survey on Deep Learning with Noisy Labels: How to Train Your Model When You Cannot Trust on the Annotations?* [online] Available at: https://arxiv.org/abs/2012.03061.
[26] Song, H., Kim, M., Park, D., Shin, Y. and Lee, J.-G., 2021. *Learning from Noisy Labels with Deep Neural Networks: A Survey*. [online] Available at: https://arxiv.org/abs/2007.08199.
[27] Patrini, G., Rozza, A., Menon, A., Nock, R. and Qu, L., 2021. *Learning from Noisy Labels with Deep Neural Networks: A Survey*. [online] Available at: https://arxiv.org/abs/ 1609.03683.
[28] Garg, S., Ramakrishnan, G. and Thumbe, V., 2021. *Towards Robustness to Label Noise in Text Classification via Noise Modeling*. [online] Available at: https://arxiv.org/abs/2101.11214.
[29] Wang, H., Lin, B., Li, C., Yang, Y. and Li, T., 2011. *Learning with Noisy Labels for Sentence-Level Sentiment Classification*. [online] Available at: https://arxiv.org/abs/1909.00124.
[30] Malik, H.H. and Bhardwaj, V.S., 2011. Automatic training data cleaning for text classification. *2011 IEEE 11th International Conference on Data Mining Workshops*, Vancouver, Canada, December 11, 2011, pp. 442-449.
[31] Eamwiwat, C., Thanasutives, P., Saetia, C. and Chalothorn, T., 2019. Using label noise filtering and ensemble method for sentiment analysis on Thai social data. *The 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing*, Chiang Mai, Thailand, October 30-November 1, 2019, pp. 251-256.
[32] Yu, H. and Hatzivassiloglou, V., 2003. Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Sapporo, Japan, July 11-12, 2003, pp. 129-136.
[33] Liu, Y., Yu, X., Liu, B. and Chen, Z., 2014. Sentence-level sentiment analysis in the presence of modalities. *15th International Conference on Intelligent Text Processing and Computational Linguistics*, Kathmandu, Nepal, April 6-12, 2014, pp. 1-16.

[34] Prasetiyowati, M.I., Maulidevi, N.U. and Surendro, K., 2021. Determining threshold value on information gain feature selection to increase speed and prediction accuracy of random forest. *Journal of Big Data*, 84, 2-22.
[35] Polpinij, J. and Luaphol, B., 2021. Comparing of multi-class text classification methods for automatic ratings of consumer reviews. *International Conference on Multi-disciplinary Trends in Artificial Intelligence*, Kuala Lumpur, Malaysia, November 17-19, 2021, pp. 164-175.
[36] Huq, M.R., Ali, A., and Rahman, A., 2017. Sentiment analysis on Twitter data using KNN and SVM. *International Journal of Advanced Computer Science and Applications*, 8(6), 9-25.
[37] Shakhovska, K., Shakhovska, N., and Veselý, P., 2020. The sentiment analysis model of services providers' feedback. *Electronics*, 9(11), 2-15.
[38] Polpinij, J., and Ghose, A.K., 2008. An ontology-based sentiment classification methodology for online consumer reviews. *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Washington DC, USA, December 9-12, 2008, pp. 518-524.
[39] Luaphol, B., Polpinij, J., and Kaenampornpun, M., 2021. Mining bug report repositories to identify significant information for software bug fixing. *Applied Science and Engineering Progress*, 15(3), DOI: 10.14416/j.asep.2021.03.005.
[40] Pennington, J., Socher, R., and Manning, C., 2014. Glove : Global vectors for word representation. *The Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Doha, Qatar, October 25-29, 2014, pp. 1532-1543.