Hindawi
Complexity
Volume 2022, Article ID 3188449, 14 pages
https://doi.org/10.1155/2022/3188449

Research Article

Vietnamese Sentiment Analysis under Limited Training Data Based on Deep Neural Networks

Huu-Thanh Duong,[1] **Tram-Anh Nguyen-Thi**,[2] **and Vinh Truong Hoang**[1]

[1]*Faculty of Information Technology, Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam*
[2]*Department of Fundamental Studies, Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam*

Correspondence should be addressed to Vinh Truong Hoang; vinh.th@ou.edu.vn

Received 26 February 2022; Revised 9 May 2022; Accepted 24 May 2022; Published 30 June 2022

Academic Editor: Manman Yuan

The annotated dataset is an essential requirement to develop an artificial intelligence (AI) system effectively and expect the generalization of the predictive models and to avoid overfitting. Lack of the training data is a big barrier so that AI systems can broaden in several domains which have no or missing training data. Building these datasets is a tedious and expensive task and depends on the domains and languages. This is especially a big challenge for low-resource languages. In this paper, we experiment and evaluate many various approaches on sentiment analysis problems so that they can still obtain high performances under limited training data. This paper uses the preprocessing techniques to clean and normalize the data and generate the new samples from the limited training dataset based on many text augmentation techniques such as lexicon substitution, sentence shuffling, back translation, syntax-tree transformation, and embedding mixup. Several experiments have been performed for both well-known machine learning-based classifiers and deep learning models. We compare, analyze, and evaluate the results to indicate the advantage and disadvantage points of the techniques for each approach. The experimental results show that the data augmentation techniques enhance the accuracy of the predictive models; this promises that smart systems can be applied widely in several domains under limited training data.

1. Introduction

The rapid growth of Internet and web technologies has motivated the development of smart systems to automate many tedious and expensive duplicated activities of human beings. Many achievements have been obtained in computer vision, natural language processing (NLP), and so forth to resolve several problems and increase the labor performances. One of the problems in NLP has attracted the attention of many research teams, namely, sentiment analysis.

In the past, the customers referred to their friends or relatives' reviews to determine whether to buy a product. Today, they obtain the reviews of any products more quickly and completely from the digital communities which appear more and more. Not only the customers but also the traders rely on these reviews to exploit the customers' opinions and expectations and understand the market trends more easily. Furthermore, they can make decisions and determine their strategies and directions to develop the products and businesses. However, if this is performed manually, this will be a tedious, time-consuming, and costly task. Sentiment analysis (SA) is a good solution for this problem.

SA indicates the degrees of the customers' pleasantness for the bought products without user intervention. The original problem has grouped the data into two classes as positive and negative polarities. SA helps to not only save time and money but also reach the customers' expectations as rapidly as possible. There are several approaches which are lexicon-based, machine learning-based, and hybrid-based (combining lexicon-based and machine learning-based). The most recent approach is deep learning model which learns the multiple layers of the feature representations of the training data to build the predictive model.

The lexicon-based approach has relied on the emotional lexicons, grammatical structure, and syntactic patterns to determine the sentiments in text. This approach needs many

related resources which depend on various languages and contexts such as sentiment wordnets, as well as a set of emoji icons. Many preprocessing techniques, which normalize the text and remove some redundant information, have usually been applied in this approach.

Obviously, SA is one of classification problems and the machine learning algorithms are easy to apply. There are many studies using supervised, semisupervised, and unsupervised learning for this problem. The supervised learning uses the well-known classifiers such as naive Bayes, logistic regression, random forest, and SVM (support vector machine) or some ensembles of various classifiers such as OVO (One versus One) and OVA (One versus All). Feature selection is also investigated to improve the performances of the classifier in this approach. This aims to retain the useful features and ignore the redundant features. The main drawback mainly depends on the size and quality of the annotated datasets in the specified domain. This is actually a challenge for low-resource languages and building the annotated datasets is expensive as regards both human resources and money. Thus, many semisupervised approaches gradually appear; these methods leverage the strong points of the lexicon-based approach and also investigate the impacts of preprocessing techniques. This one also motivates the appearance of text augmentation techniques to enrich data training from the original training data to raise performances of the classifiers.

Due to the rapid development of computing power, deep learning (DL) obtains the state-of-the-art performances in computer vision and natural language processing (NLP). This learns multiple layers of the features and representations and has millions of parameters, thus requiring a large amount of the training data in order to avoid overfitting and obtain the generalization of the predictive models. DL has appeared as a form of artificial neural networks since the 1990s and has only been trained with one layer or two layers (calling shallow neural networks). According to Zhang et al. [1], the rapid growth of DL (deep neural networks) has been recently as follows: (1) the availability of computing power due to the advances in hardware, (2) the availability of huge amounts of training data, and (3) the power and flexibility of learning intermediate representations. The authors in [1] also summarized many methods based on deep learning for sentiment analysis problems. Karuppusamy [2] presented the analysis of neural networks in the modeling of the language. Language model is a fundamental process which is applied for many problems such as sentence completion, translation of the statistical machines, and text generation. The author showed that the neural network is a good choice to obtain quality language models. Moreover, many deep learning models have been invented to improve the performances of the prediction such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Recursive Neural Network (RecNN) [1], as well as many improved modern models [3, 4]. DL has also been investigated in various domains. Zhou et al. [5] applied DL for fault diagnosis of industrial machinery. To achieve trustworthy diagnosis, they conducted the fault diagnosis in a probabilistic Bayesian deep learning framework (proposed by Caceres et al. [6]) by exploring an uncertainty-aware model to understand the unknown fault information and identify the inputs from unseen domains.

Lack of the training data is one of the big challenges to build a powered-AI system and a main barrier to broaden AI in many fields whose datasets are unavailable and cost a lot to build. This may lead the models to less generalization, and they may be prone to overfit training data and result in an uncertain prediction. DL models have required a large amount of training data to enhance the model robustness and reduce overfitting. Data augmentation has emerged as a spotlight approach; this has been inspired by image classification to enhance the abundance of the training data. It has been equipped in NLP and achieved promising performances. Unlike image classification, where there are handcrafted rules (such as scale, rotate, noise, and mixup), NLP faces many challenges due to the complexity of the languages and it is difficult to determine the rules to generate data.

Our main contribution is to implement the methodologies which can boost the performances of the predictive models under limited training data. In addition to summarizing the approaches for the effective preprocessing techniques and text augmentation, we also propose some relevant methods in preprocessing techniques and ensembles of text augmentation techniques. Many various experiments are conducted for Vietnamese sentiment analysis problems to analyze and evaluate the advantage and disadvantage points of the experimented results to indicate the effective methods in each approach. The predictive models consist of the well-known machine learning-based classifiers and deep learning models.

The rest of this paper summarizes the related works such as preprocessing techniques, text augmentation techniques, and the effective models for sentiment analysis problems in Section 2. Section 3 presents the methods which we use to experiment, analyze, and evaluate the performances. Section 4 shows several experiments and their discussions. The conclusions are presented in Section 5.

2. Related Works

The recent studies concern the user-generated text on social networks, blogs, e-commerce websites, or some reviewing websites for foods, films, watches, travels, and so forth. Therefore, they may contain much unnecessary information; this not only affects the accuracy results of the models but also makes the redundant computational cost increase. The preprocessing techniques are crucial tasks for almost all problems, especially for lexicon-based and semisupervised approaches; they help to focus on the necessary information and reduce the complexity of computation. Singh et al. [7], Jianqiang et al. [8], and Symeonidis et al. [9] performed several experiments to evaluate the impacts of the preprocessing techniques; the authors presented that they significantly boosted the performance of the models. Common techniques are usually used such as lowercasing, removing stop-words, punctuations, numbers, URLs, hashtags, user mentions, or replacing slang words by actual

words, acronyms, negative forms, emoji icons, or handling intensification terms, as well as concessive clauses.

For sentiment analysis problems, some techniques are especially essential to determine the emotions in the texts, namely, negation handling, emoji icons handling, and spelling correction. The negation forms can be ambiguous of the classifying text and entrap the algorithms. Ghag et al. [10] increased slightly the performance of sentiment analysis after negation handling; they ignored the negation terms, but the next term was recorded in the term document vector of opposite orientation. Al-Sharuee et al. [11] built an antonym dictionary based on SentiWordNet to replace the adjectives and adverbs following the negation terms with their opposite sentiment words. These studies mostly concerned the next term of the negation modifier. However, the negation modifier not only impacted the next term but also extended other terms following negation. Farooq and Umar [12] improved the performance of sentence level sentiment analysis in the presence of negation by identifying the scope of negation while determining the sentence polarities. They indicated two types of negation as morphological and syntactic negations separately: for morphological negation a set of the prefix and suffix were used to handle and for syntactic negation the linguistic features (conjunction analysis, punctuation marks, and heuristics based on POS of negation term) were used to determine the scope of affectation.

Emoji icons are booming in digital communications; they are used as shorthand to express ideas, give minimal information faster than writing, and carry many sentiments. Many studies investigate their important roles for both lexicon-based and machine learning-based approaches. Hogenboom et al. [13] analyzed and exploited emotions to significantly improve a state-of-the-art lexicon-based sentiment classification method. Wang et al. [14] analyzed the relationship between emotional icons and sentiment polarities; they indicated that the emotional icons are strong signals to convey the sentiment polarities. Kralj Novak et al. [15] proposed Emoji Sentiment Ranking for automated sentiment analysis; the analyses showed the important roles of the emoji icons for sentiment distribution in tweet and there are no significant differences in the emoji icons rankings between 13 European languages. Guibon et al. [16] introduced approaches exploiting emoji icons in sentiment analysis.

Wrong spelling occurs very often in the unstructured texts and increases the computational complexity due to the redundant dimensions. Kim [17] used AutoMap to correct the common spellings. Moreover, unknown words appear more and more in social communications; they affect the performance of the predictive models. Liu et al. [18] suggested the contextual substitution based on the semantic similarity of word2vec model to process the unknown words in financial public opinions. They obtained a better classification than the traditional methods of identifying the categories of financial public opinions.

Feature selection is also an interesting approach to select the most relevant features. This reduces the redundant features, minimizing the cost and time for building the prediction models. There are three types of feature selection, namely, filter (select features based on the statistical tests such as variance and chi-square), wrapper (select features by using trained models), and embedded methods (select features during the model training). Feature extraction obtains the meaningful low-dimensional representation of the high-dimensional data. Kim [17] built the process to reduce the dimensionality by using feature extraction while preserving the advantages and overcoming the drawbacks of feature extraction for sentiment analysis problems.

Data augmentation enriches the training data to increase the generalization of the prediction models and overfitting avoidance. This aims to generate new samples having the same label as the original data. Abulaish and Sah [19] improved the performance of CNN relying on text augmentation which combined n-grams and Latent Dirichlet Allocation (LDA) to identify class-specific phrases to enrich the data. There are many other approaches carried out in NLP effectively such as lexicon substitution, sentence shuffling, contextual substitution, back translation, syntax-tree transformation, and embedding mixup.

Lexicon substitution replaces a lexicon by the synonym lexicons based on wordnets or thesaurus. This is the most simple approach, which is easy to implement and maintains the meaning of the text. However, it needs effective wordnets or thesaurus; this is easy to obtain for high-resource languages such as English using wordnets of NLTK lib (https://www.nltk.org/), but the low-resource languages are a big challenge. Other lexicon-based techniques are commonly used such as shuffling the lexicons, randomly inserting a synonym lexicon, randomly swapping two lexicons, and deleting the lexicons in text with a probability. These ones do not need any additional resources to implement and still achieve a promising performance although they might generate meaningless text. Wei and Zou [20] demonstrated the effectiveness of these techniques; their experiments included replacing n words by the synonym words randomly, swapping two random words, deleting a random word with a probability, and inserting the synonym words of a word randomly.

Back translation is a successful method that improves the performances of the machine translation (Sennrich et al. [21], Fadaee et al. [22], Sugiyama and Yoshinaga [23], and Xia et al. [24]). When translating the text between source and destination languages, it is not totally similar to the original text after back translating but still retains the meaning. This feature is leveraged, as well as the abundance of the high-resource languages such as English to generate new training data. This technique can be applied easily in any language and bring good performances. However, it needs an effective translator, especially for working with free text, which is without any constraints, and wrong spellings and other noise features occur, which is also a challenge for this approach.

Syntax-tree transformation utilizes a syntax tree and applies some syntactic grammars to obtain the transformed tree which is used to generate the new text. This approach is a good technique to boost the performance, but it is expensive in calculation because of the complexity of the language

features. Word embedding is a technique to convert a text into vectors (character, word, sentence, and document levels). The words which have a similar context will be near each other in vector space. Fortunately, the enormous resources on the Internet and outstanding development of pretrained models provide many available word-embedding models based on word2vec, GloVe, BERT, and so forth. This has opened many directions to leverage the word-embedding features in NLP.

Naturally, this refers to a heuristic approach for contextual substitution with the hope that the words which have the same meaning will also be near each other on embedding space. This approach will replace a word by one of its contextual words by using the pretrained word embeddings such as word2vec, GloVe, and BERT. This reduces the cost to build the relevant resources such as wordnets or thesaurus. However, due to being a black-box and context-based method, the opposite words might sometimes be near each other in the vector space, such as "tốt" (good) and "xấu" (bad); this is able to be harmful for the training model. Kobayashi [25] proposed the contextual augmentation by using bidirectional language model to replace a word with other words. Kobayashi improved a language model with a label-conditional architecture to maintain the label compatibility.

Masked language model (MLM) is a method to fill a blank in a sentence with suitable words based on the contexts. This may be leveraged to generate new samples by using the pretrained models with BERT (Bidirectional Encoder Representations from Transformers) and leave some masks in the text and ask BERT to predict the words for that mask.

The most recent embeddings mixup emerged to use image augmentation by merging two random images in a proportion to generate the synthetic samples. Guo et al. [26] proposed two strategies for mixup on sentence classification by performing the interpolation on word embeddings and on sentence embeddings, called wordMixup and senMixup. The wordMixup approach conducts sample interpolation in the word embedding and the senMixup approach is conducted on the mixup layers before feeding to softmax layers to produce the predictive model. Sun et al. [27] stacked a mixup layer over the final hidden layer of the pretrained transformer-based model and explored the effectiveness of mixup in the transformer. They promoted the performance across a wide range of NLP benchmarks. Ding et al. [28] proposed a novel approach with a language model for sequence tagging tasks. Firstly, they linearize the tagging sentences and then use a language model to learn the distribution of the words and their tags. This model was used to generate labeled data. This is an effective method for both semi-supervised and unsupervised approaches and no need the additional resources.

Moreover, Jason Wei et al. [29] proposed the multitask view approach. Most of the previous studies of data augmentation aimed to generate new samples that resemble the original sample for prediction models; their proposal will train the original samples and the augmented samples separately, and this approach will join these tasks by using a weighted cost function so that both original and augmented data receive substantial weight during training.

3. The Methodologies

3.1. The Preprocessing Techniques. Our study focuses on the unstructured texts which are informal and free text and do not comply with any constraints. Thus, they contain a lot of noise information such as wrong spelling, wrong grammar, and unknown words. Preprocessing techniques are used mostly in AI in general and NLP in particular. This section presents the popular preprocessing techniques and our approach uses these techniques for Vietnamese sentiment analysis.

Several experiments are performed to assess the suitability and effectiveness of each preprocessing technique for our problem. Furthermore, in order to obtain good quality of the generated data, the original data such as elongated characters removal, punctuation removal, and emoji icons substitution have to be normalized. These ones also help to improve the performances and reduce the redundant features and minimize the cost and time to build the predictive models. To prove their impacts on the quality of augmentation data, we also perform the experiments combining preprocessing techniques and data augmentation techniques in the experimental section and obtain promising results.

The popular techniques that are always used include Vietnamese tokenization using pyvi (https://pypi.org/project/pyvi/) ("Tôi hài lòng điện thoại này" (I'm pleased with this phone) converts to "Tôi hài lòng điện_thoại này"), lowercasing (such as "Hài_lòng" converts to "hài_lòng"), and stop-words removal (such as "thì" and "là"). The remaining techniques are used separately such as elongated characters, emoji icons substitution, negation handling, number removal, and punctuation removal to evaluate and choose the effective methods for Vietnamese sentiment analysis. These techniques increase the semantics of the text and reduce the dimensions in vector space.

Using the elongated characters in a word becomes a habit for social communications and aims to emphasize some emotions. These words will be removed and only one character will be retained, respectively; for example, "tuyệt_vờiiiiiii" (great) and "đẹp quuuuuuuuá" (so beautiful) are converted to "tuyệt_vời" and "đẹp quá." Number removal is the removal of numbers which express less emotions in the text, but some numbers can bring the sentiment via emoji icons such as 3>, so this should be performed after emoji icons handling. The punctuations usually do not affect the emotion in the text; they should be removed to reduce the dimensions in vector space.

POS tagging, which is a crucial tool in NLP, assigns the part of speech to each word in a sentence. This technique only retains the words whose part of speech may contain sentiments, including noun, verb, pronoun, adverb, and adjective. Symeonidis et al. [9] kept nouns, verbs, and adverbs for their experiments. For unknown words or some wrong spellings, we replace the contextual word based on the MLM.

In order to exploit the emoji icons, we incorporate a set of emoji icons built by [15] with our emoji icons built manually. Based on these sets, we replace the positive and negative icons by appending the "positive" and "negative" lexicons, respectively. For example, consider the sentence "Quá đẹp! Apple chưa bao giờ làm tôi thất_vọng:)" (So beautiful! Apple has never made me disappointed:)!), where the ":)" is a positive emoji; the result is "Quá đẹp! Apple chưa bao giờ làm tôi thất_vọng *positive*!".

The negation form contains ambiguities and can easily lead to misunderstanding of the classifiers. For example, the sentence "điện_thoại chẳng đẹp, tôi không hài_lòng" (this phone is not beautiful, I'm unpleasant) is a negative sentiment, but if it does not concern the negative terms "không" (not) and "chẳng" (not), then the classifiers have the positive result. The negation forms are detected by using syntactic features; some modifiers usually present the negation form such as "không" (not), "chẳng" (not), "chưa" (not yet), "không được" (not), "không biết" (unknown), and "chả" (not). We provide two methods to handle the negation. In the first one, when it detects a negation modifier followed by a positive or negative lexicon, the modifier and its next lexicon will be replaced by "notpositive" or "notnegative" lexicons, respectively. The second one uses our pretrained model built by word2vec; a negation modifier and its next lexicon will be replaced by a word having the opposite orientation. Ideally, it may be the antonym word. For example, consider the sentence "sản_phẩm không đẹp, tôi cảm_thấy thất_vọng" (the product is not beautiful, I feel disappointed), where "không" (not) is a negation modifier and "đẹp" is a positive lexicon; they are replaced by "notpositive" lexicon with the first approach, and the result is "sản_phẩm" *notpositive*, tôi cảm_thấy thất_vọng." For the second approach, they are replaced by "xấu" (bad), and the result is "sản_phẩm xấu, tôi cảm_thấy thất_vọng" (the product is bad, I feel disappointed). The negation is an ambiguous form for many problems in NLP. Thus, handling negation not only boosts the performance of sentiment analysis but also is a buffer step to get the good generated data based on some NLP techniques such as synonym substitution, syntax-tree transformation, or back translation.

For intensifier lexicons (such as "rất" (very), "cực kỳ" (extremely), and "quá" (so)) the "strongpositive" and "strongnegative" will be appended if their next lexicon is positive or negative lexicon, respectively. For example, consider "Tôi rất ưng_ý, nó giúp công_việc của tôi trở_nên dễ_dàng hơn" (I'm very pleased, it helps my work becoming more easily); the word "ưng_ý" (pleased) is a positive lexicon and the word "rất" (very) is an intensifier lexicon; thus the result is "Tôi rất ưng_ý, nó giúp công_việc của tôi trở_nên dễ_dàng hơn *strongpositive*".

Moreover, in order to leverage the lexicon features, we utilize VnEmoLex list [30] and our manual lists of positive and negative lexicons which are used regularly. When detecting a positive or negative lexicon, it will, respectively, append "positive" and "negative" lexicons to the current text; it is certainly conducted after negation handling and intensifier handling. Back to above example of negation handling, after obtaining the sentence "sản_phẩm *notpositive*, tôi cảm_thấy thất_vọng," the word "thất_vọng" (disappointed) is a negative lexicon; thus the result is "sản_phẩm *notpositive*, tôi cảm_thấy thất_vọng *negative*."

3.2. The Text Augmentation Techniques. This section presents the text augmentation to generate new samples automatically, including synonym substitution, lexicon insertion, deletion, swapping, sentence shuffling, contextual substitution, masked language model, back translation, syntax-tree transformation, and embedding mixup.

Firstly, based on the easy ideas of Wei et al. [20], the goal is to generate new samples by replacing some random words by their synonym words, inserting the synonym words of some random words into the sentence, swapping two random words in the sentence, and deleting words with probability p. For synonym substitution and synonym insertion, we use a Vietnamese thesaurus dataset provided by viet.wordnet.vn (https://github.com/zeloru/vietnamese-wordnet—the latest access 12/05/2021) to determine synonym words. Each sample in training data will be performed 10 times for each method; a time will generate 4 new samples responding to four methods. Thus, from the 100 samples of each category, 4,000 new samples will be generated.

Another easy idea is sentence shuffling; the sentences among the paragraphs or in the same paragraph are shuffled to obtain new samples. The underthesea lib (https://underthesea.readthedocs.io/en/latest/readme.html) is used to tokenize the text into the sentences. This method will generate 1000 samples for each polarity by shuffling the sentences among the paragraphs in the same labels.

Back translation leverages the translator machines and the abundance of high-resource languages such as English. The translated text, which translates from the source language to destination language, is back-translated not being totally the same as the original text, thereby obtaining new samples. Google Translate API is used to implement a back translation approach; the text will be translated from Vietnamese to English and vice versa for the generated samples. From 100 training samples of each polarity, 100 more samples of each one will be generated.

Syntax-tree transformation is a rule-based approach which uses some syntactic grammars to transform the syntax tree to the transformed tree for new samples. Our work will generate 100 more new samples for each polarity by transforming the active voice to passive voice.

We also collect over 100 thousand of the comments or reviews in e-commerce websites to build a word2vec model to implement the contextual substitution. Word2Vec model indicates the vector of a word based on around words, called contextual words. This pretrained model is used to determine the contextual words of a random word in the text to replace new augmented samples. As presented in the above section, the contextual substitution is a black-box approach which might be harmful for the original text because the opposite words might be used for replacement; we also use $tf \times idf$ score to choose the replacing words in the text. The words are informative if they have a low $tf \times$

idf score, so replacing them does not affect or little affects the meaning of the original text. For both cases, each sample will be run 10 times to obtain 10 more samples, so it will generate 1000 samples from 100 training samples of each category.

MLM can predict a masked word in the context; this idea can be used to generate new samples. Our dataset is not big enough to build a pretrained model with BERT, so we use PhoBERT instead. Pretrained PhoBERT is a state-of-the-art language model for Vietnamese proposed by Dat Quoc Nguyen and Anh Tuan Nguyen [31]. This method will generate 200 samples for each category. Besides, we also use PhoBERT for replacing the unknown words.

Mixup is a domain-independent technique to generate synthetic samples for training; it was proposed by Zhang et al. [32] for image classification. This combines the linear interpolations of the feature representations of a pair of input images randomly and their training labels.

$$\begin{aligned}\widehat{x} &= \lambda x_i + (1-\lambda)x_j,\\ \widehat{y} &= \lambda y_i + (1-\lambda)y_j,\end{aligned} \tag{1}$$

where x_i and x_j are the raw input vectors, y_i and y_j are the corresponding one-hot labels, and λ is the mixup ratio having the value between [0, 1].

This idea also is equipped in NLP and shows a stable and effective solution which also avoids overfitting. We also consider the impacts of this approach on Vietnamese sentiment analysis by injecting the mixup layer over the final layer of the deep learning model. Furthermore, we incorporate this method with some other augmented methods such as back translation, contextual substitution, and synonym substitution and achieve competitive results.

3.3. The Machine Learning-Based Classification. Machine learning is an important part of artificial intelligence; it learns from the labeled training data and can predict the labels of the new data points. This has been categorized into supervised learning, unsupervised learning, semisupervised learning, and reinforcement learning. The supervised learning has been based on the labeled dataset to predict the label of the new data, the unsupervised learning does not need a labeled dataset, it works based on the structure of the data, and the semisupervised learning has a little bit of a labeled dataset to learn and give the prediction. Our problem has emerged as the semisupervised learning with the limited training data. The experiments use the machine learning-based predictive models; we choose the well-known classifiers and obtain the state-of-the-art performances including logistic regression, support vector machine, OVO (One versus One), and OVA (One versus All).

Logistic regression is a statistical approach used widely in prediction for both regression and classification problems. The training phase indicates the relationship between a dependent variable y and the independent variables x_i. It predicts the label of a data point based on the probability of the logistic function. For classification problems, this uses a predefined threshold belonging to [0, 1], and the value is usually 0.5. If the probability of a data point is greater than or equal to the threshold, then it belongs to the positive class; otherwise, it belongs to the negative class. The logistic function is as follows:

$$f(x) = \text{logistic\_func}\left(w^T x + b\right). \tag{2}$$

This needs to identify the coefficients w^T and b based on the training data points. The logistic function is usually used as a sigmoid function which is a continuous and differentiable function.

Support vector machine (SVM) is a strong classifier which finds a boundary (hyperland) between the training data points belonging to different categories. A good hyperland obtains the maximum of the margins which are the distances between the hyperland and the closest data points from each category. This classifier can work for both linear and nonlinear datasets.

The cost function of linearly separable dataset is

$$f(x) = w^T x + b. \tag{3}$$

This needs to identify w and b satisfying

$$(w, b) = \text{argmin}_{w,b}\frac{\|w\|_2^2}{2}. \tag{4}$$

For linearly separable dataset with any noise data points, it needs to identify w and b satisfying

$$(w, b) = \text{argmin}_{w,b,\xi}\frac{\|w\|_2^2}{2} + C\sum_{i-1}^{n}\xi_i, \tag{5}$$

where $y_i(w^T x_n + b) \geq 1$, $\forall\ i$ are indexes of data points, C is the cost constant, and ξ is slack variable.

For nonlinearly separable dataset, SVM uses the kernel functions to transform them to linear space. In this paper, we use the RBF (Radial Basis Function) kernel.

OVO and OVA are ensembles of binary classifiers for multiclass classification, where OVA performs c iterations (c is the number of labels), each i-th iteration ($1 \leq i \leq c$) indicates whether a data point belongs to c_i, and the final prediction is determined by a probability. OVO performs $c(c-1)/2$ iterations, each iteration takes pairwise labels and indicates which label a data point belongs to, and the final prediction is determined based on the major voting of the iterations.

3.4. The Deep Learning-Based Models. Deep learning, which is a powerful machine learning technique, learns multiple layers of representations and features of the data. It consists of one input layer, one output layer, and many hidden layers between input and output layers. The lower layers learn the simple features and the higher layers learn more complex features from the outputs of the lower layers.

Figure 1 is a neural network with three layers (two hidden layers and one output layer, excluding the input layer). The circle in the input layer represents an element of

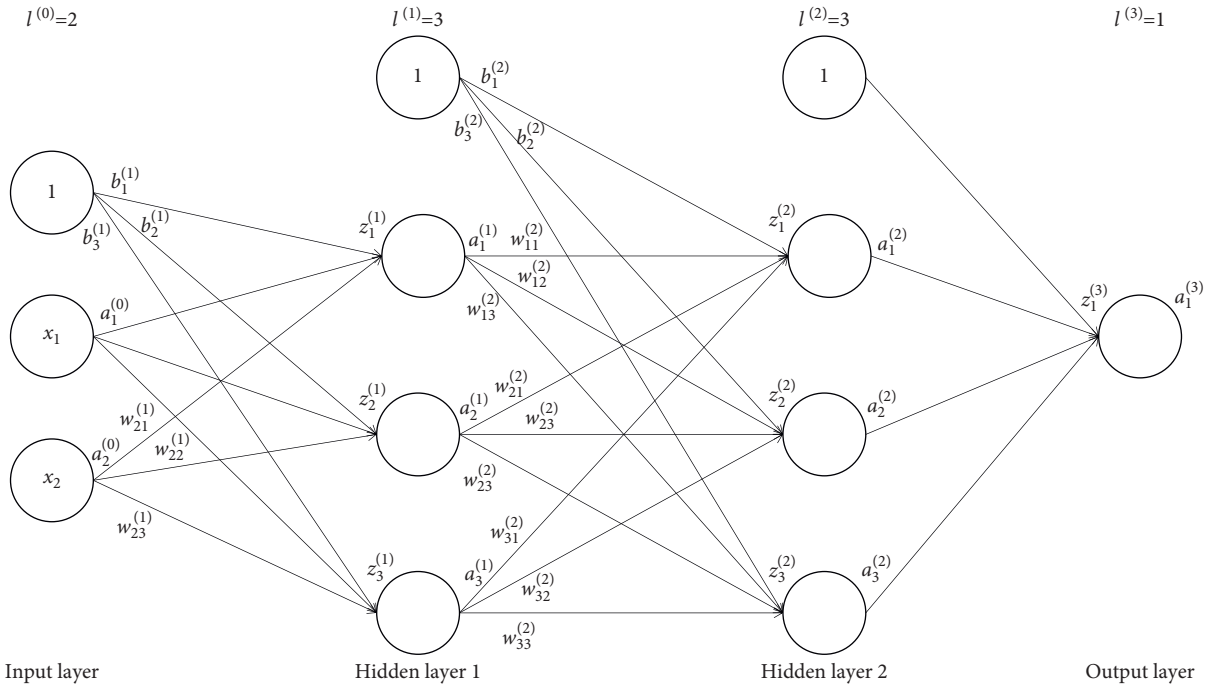

Figure 1: The neural network has the input layer and three layers (two hidden layers and the output layer).

the input vector and the circles in the hidden and output layers represent the neurons. A layer consists of a large number of these circles, called nodes (units) commonly. The connection line between two neurons of two near layers associates a weight. The neurons in a layer will read the output from the neurons of the previous layers, process the information, and return output to the neurons in the next layers. The outputs of the hidden layers are achieved by using activation functions which are usually nonlinear functions such as sigmoid function, tanh function, and ReLU function. The neurons of the output layers use the softmax functions.

Put $l^{(k)}$ is the number of nodes in a layer (not including bias node); $W^{(k)}$ is the weight matrix between (k-1)-th layer and k-th layer, where $w_{ij}^{(k)}$ is the weight between the i-th node in (k-1)-th layer and the j-th node in the k-th layer; $b_i^{(k)}$ is the bias weight of the i-th node in the k-th layer, δ is an activation function.

$$z^{(k)} = \left(W^{(k)}\right)^T * a^{(k-1)} + b^{(k)},$$
$$a^{(k)} = \delta\left(z^{(k)}\right). \tag{6}$$

The above process is called feedforward, after this step saves the activations $a^{(l)}$. Training the deep learning models uses stochastic gradient descent based on backpropagation which calculates the gradients from the output layer to the first layer.

The deep learning models have emerged as the state-of-the-art performances in computer visions and recently in NLP. First is to learn the word vector representations using neural networks models. Word2Vector is one of the most popular approaches to transform a word to a vector; it calculates from a very large dataset and uses a neural network consisting of one input layer, one output layer, and one hidden layer. There are two methods: skip-gram and CBOW (Continuous Bag of Word); skip-gram predicts a word based on its context, and CBOW predicts a context based on its words. In this paper, we have built a word2vector model for Vietnamese reviews and used it to augment new data via contextual substitution for sentiment analysis problems.

Another method learns the vector representations as an autoencoder which is a three-layer neural network; it learns new representations from input representations. Firstly, the input vector (one-hot vector) is mapped to the hidden layer via encoder function, and the representations of the hidden layer are mapped to the output layer via decoder function to obtain new representations. Because encoder and decoder functions are nonlinear functions, the autoencoder can learn nonlinear representation. Thus, the new representations are better expressed than the original representations.

Araque et al. [3] presented some uses of deep learning for sentiment analysis such as augmenting the knowledge contained in the embedding vectors with other sources of information. This information can be sentiment specific word embedding or a combination of hand-crafted features and these sentiment specific word embeddings. Another approach is to incorporate new information to the embeddings, where deep learning has been used to extract sentiment features combining with semantic features.

4. The Experiments

This section evaluates the effectiveness of each technique of preprocessing and text augmentation. The experiments use the datasets mentioned in our paper [33] consisting of 4 datasets (Table 1 shows the size of the datasets). Figure 2 shows the main flowchart to perform the experiments. It

Table 1: The size of the datasets.

No.	Datasets	Polarities	Size
1	Dataset 1	Positive	9,280
		Negative	6,870
2	Dataset 2	Strong positive	2,380
		Positive	2,440
		Negative	2,380
3	Dataset 3	Positive	15,000
		Negative	15,000
4	Dataset 4	Positive	5,000
		Negative	5,000

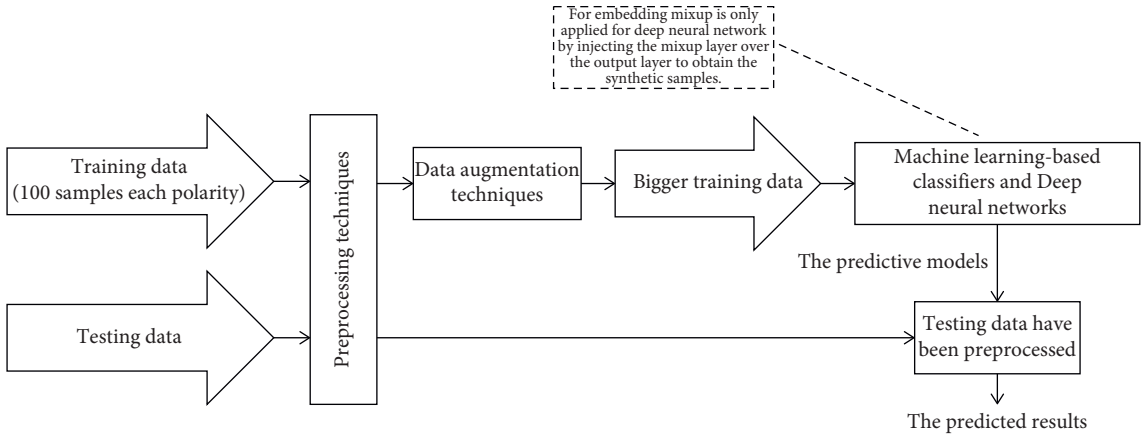

Figure 2: The experimental flowchart.

Table 2: The accuracy results of various preprocessing techniques for Vietnamese sentiment analysis based on machine learning classifiers.

Datasets	Classifiers	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Dataset 1	LR	0.822	0.823	0.822	0.823	0.828	0.823	0.843	0.852	0.829
	SVM	**0.832**	**0.833**	**0.832**	**0.833**	**0.830**	**0.835**	**0.852**	**0.856**	**0.837**
	OVO	0.826	0.827	0.826	0.826	0.829	0.828	0.848	0.852	0.833
	OVR	0.826	0.827	0.826	0.826	0.829	0.828	0.848	0.852	0.833
Dataset 2	LR	0.707	0.707	0.707	0.707	0.703	0.712	0.740	0.742	0.707
	SVM	0.673	0.672	0.673	0.673	0.664	0.686	0.707	0.698	0.673
	OVO	**0.715**	**0.715**	**0.715**	**0.715**	**0.705**	**0.718**	**0.748**	**0.745**	**0.714**
	OVR	0.697	0.699	0.697	0.698	0.696	0.708	0.735	0.738	0.698
Dataset 3	LR	0.798	0.797	0.798	0.798	0.790	0.806	**0.826**	0.802	0.800
	SVM	0.799	0.797	0.799	0.800	0.789	0.806	0.822	0.811	0.801
	OVO	**0.800**	**0.799**	**0.800**	**0.801**	**0.792**	**0.807**	0.822	**0.813**	**0.803**
	OVR	**0.800**	**0.799**	**0.800**	**0.801**	**0.792**	**0.807**	0.822	**0.813**	**0.803**
Dataset 4	LR	0.805	0.806	0.805	0.805	0.798	0.808	0.828	0.815	0.811
	SVM	**0.808**	**0.810**	**0.808**	**0.810**	**0.804**	**0.813**	**0.830**	0.822	0.812
	OVO	0.806	0.807	0.806	0.806	**0.804**	0.812	**0.830**	**0.824**	**0.813**
	OVR	0.806	0.807	0.806	0.806	**0.804**	0.812	**0.830**	**0.824**	**0.813**

(1) Without preprocessing (the baseline result); (2) number removal; (3) punctuation removal; (4) elongated characters removal; (5) POS tagging selection; (6) intensifier handling; (7) negation handling (replacing the lexicons); (8) negation handling (using pretrained models); (9) emoji icons substitution.

only uses 100 samples (about 1 to 3 percent of training data) for each polarity of the datasets for training. All data are preprocessed and generate new samples based on the data augmentation techniques. The synthetic data will be used to build the predictive models with machine learning-based classifiers or deep neural networks. Particularly, embedding mixup is only applied for deep neural networks by injecting the mixup layer over the output layer to obtain the synthetic samples.

We conduct the experiments separately to assess the effectiveness of the preprocessing techniques to choose the suitable techniques applied to our problems. Table 2 presents the accuracy results of the experimented preprocessing techniques separately, including number removal,

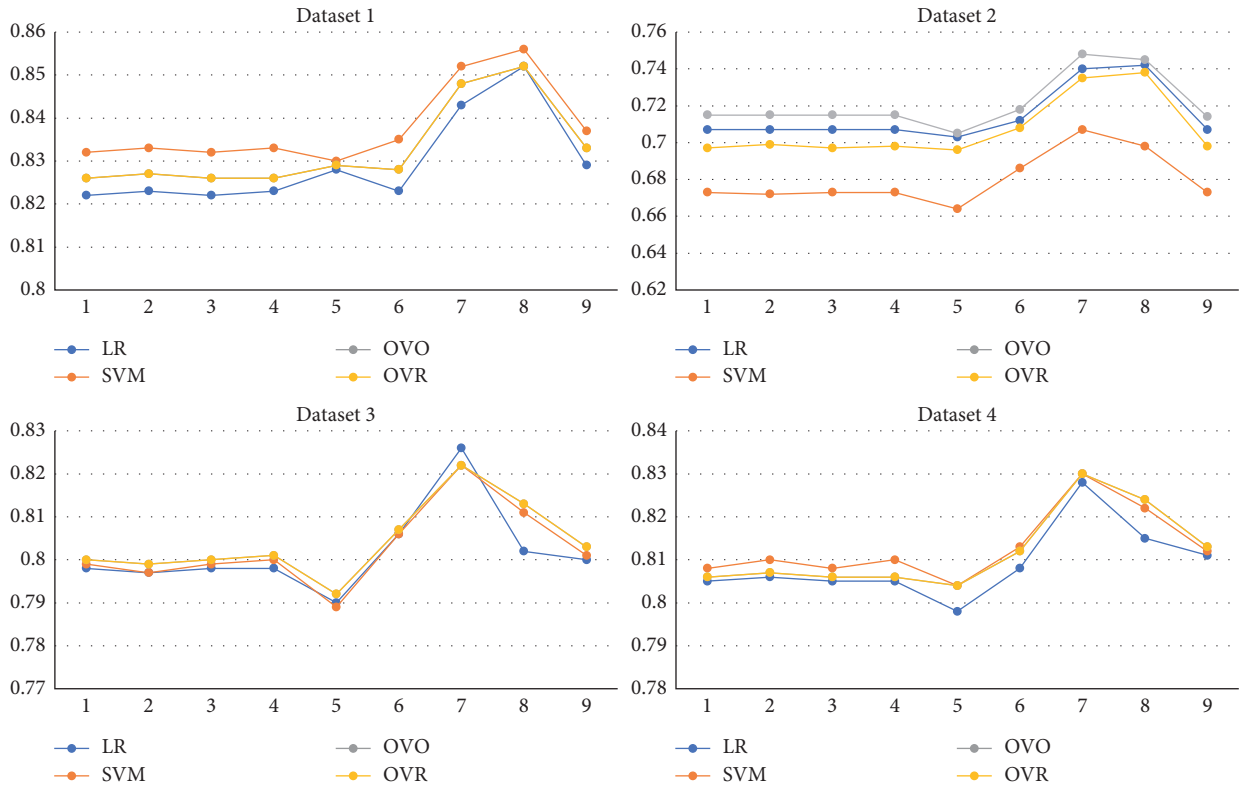

Figure 3: The accuracy results of various preprocessing techniques for Vietnamese sentiment analysis based on machine learning classifiers.

punctuation removal, POS tagging selection, intensifier handling, negation handling, and emoji icons substitution. The predictive models consist of both the well-known classifiers such as logistic regression, SVM, OVO, and OVA and deep learning models.

Based on our preceding works [34] we performed the experiments using various parameters of these classifiers, so SVM classifier uses the parameters consisting of kernel = rbf, $C = 1e5$, and gamma = 1/(number of vector features) corresponding to gamma = auto in scikit-learn lib. About the logistic regression classifier, it uses the parameters consisting of multi_class = multinomial and solver = lbfgs. The deep learning models contain two hidden layers (using ReLU activation function and 16 hidden units of each one) and the output layer (using sigmoid activation function). The loss function is binary_crossentropy function because of the probability output. The training phase is performed in 20 epochs in mini batches of 512 samples.

Figure 3 is the chart of the accuracy results of various preprocessing techniques for Vietnamese sentiment analysis based on machine learning classifiers. Number removal and POS tagging selection improve accuracy a little bit, but it is unstable in the whole datasets. The performance of the POS tagging problem depends on the grammar structure of the text; working with free text like our sentiment analysis problem is a big challenge. The accuracy of punctuation removal is mostly unchanged, only changing a little bit if we get above four decimal places in the results. Besides, it affects the semantics of the text and is harmful for some augmented techniques such as the back translation technique; this leads to some poor quality of some translated text. Thus, we do not choose these techniques for the next steps.

The accuracy results of elongated characters removal, intensifier handling, and emoji icons substitution techniques mostly improve the performances compared to the baseline results (the first column). Similarly, two approaches for negation handling also boost the performances for the datasets. The approach, which replaces the fixed lexicons, does not concern the meaning of the lexicons, but this is simple and easy to implement, does not need any additional resources, and obtains good results. The approach, which replaces the contextual words based on MLM, also achieves better accuracy results than the baseline result and overcomes the performances of a few datasets (dataset 1, logistic regression, and OVR classifiers of dataset 2), but it depends on a pretrained model and performs slowly.

In order to prove the impacts of preprocessing techniques on the quality of the augmented data, we perform the experiments by combining them with data augmentation techniques. The first column of Table 3 combines these effective methods: elongated characters removal, intensifier handling, emoji icons substitution, and negation handling. It is better than the baseline result and the single negation handling. We choose these techniques and incorporate the data augmentation techniques. Table 4 shows the size of new samples for each polarity when applying the data augmentations; Table 3 presents the accuracy results of data augmentation techniques based on machine learning classifiers and Figure 4 is their chart. Table 5 presents the

Table 3: The accuracy results of various data augmentation techniques for Vietnamese sentiment analysis based on machine learning classifiers.

Datasets	Classifiers	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Dataset 1	LR	0.850	**0.864**	0.848	**0.865**	0.863	**0.856**	0.858	**0.858**
	SVM	**0.859**	0.852	0.812	0.860	**0.865**	**0.856**	0.861	0.831
	OVO	0.854	0.857	**0.854**	0.863	0.864	**0.856**	**0.862**	**0.858**
	OVR	0.854	0.857	**0.854**	0.863	0.864	**0.856**	**0.862**	**0.858**
Dataset 2	LR	0.742	0.743	**0.752**	0.754	0.752	**0.751**	0.749	0.748
	SVM	0.721	0.741	0.711	0.754	0.736	0.746	0.720	0.732
	OVO	**0.751**	**0.745**	0.745	**0.756**	**0.759**	0.747	**0.751**	**0.750**
	OVR	0.734	0.731	0.738	0.741	0.747	0.738	0.739	0.738
Dataset 3	LR	**0.826**	**0.831**	**0.832**	**0.838**	**0.831**	**0.825**	**0.828**	**0.826**
	SVM	0.824	0.829	0.825	0.832	**0.831**	0.820	0.826	**0.826**
	OVO	0.823	0.829	0.827	0.831	0.829	0.819	0.825	**0.826**
	OVR	0.823	0.829	0.827	0.831	0.829	0.819	0.825	**0.826**
Dataset 4	LR	0.816	**0.830**	0.829	0.829	0.820	0.814	0.822	0.821
	SVM	**0.828**	0.826	**0.833**	**0.831**	**0.834**	**0.817**	**0.826**	**0.827**
	OVO	0.826	0.826	**0.833**	0.829	**0.834**	0.816	**0.826**	**0.827**
	OVR	0.826	0.826	**0.833**	0.829	**0.834**	0.816	**0.826**	**0.827**

(1) Preprocessing techniques; (2) EDA; (3) sentence shuffling; (4) back translation; (5) syntax-tree transformation; (6) contextual substitution (w2v); (7) contextual substitution ($W2V + tf \times idf$); (8) masked language model (PhoBERT).

Table 4: The size of new samples for each polarity when applying the data augmentation techniques.

No.	Data augmentation techniques	Size
1	Vietnamese EDA	4,000
2	Sentence shuffling	1,000
3	Back translation	100
4	Syntax-tree transformation	100
5	Contextual substitution ($W2V$)	1,000
6	Contextual substitution ($W2V + tf \times idf$)	1,000
7	Masked language model	200

accuracy results of data augmentation techniques with standard deviation based on deep neural networks and Figure 5 is their chart.

The first experiment of text augmentation is to replace synonym words, insert the synonym words, swap two words randomly, and delete words with probability p (called EDA approach [20]). These methods are simple to perform, with low cost in computation. Although some methods (inserting, swapping, or deleting) might produce the meaningless samples, most of algorithms for the datasets still obtain better results for machine learning-based approach and steadily improve for deep learning-based approaches (the second columns of Tables 3 and 5, respectively) compared to the baseline results (the first columns of each table). Synonym replacement is a natural idea, is easy to implement, and still retains the original meaning of the text. The biggest burden is to need an effective and sufficient thesaurus, as well as depending on the domain context and languages. A word may have many synonym words and choosing one of them affects the quality of augmented text and the predictive models; for example, the word "hai_long" has the synonym words "thoa_man," "ưng_y," "bang_long," and "man_nguyen" for the sentence "toi hai_long chat_luong sản_pham do" and replacing "hai_long" by "ưng_y" will be better than replacement by "man_nguyen" or "thoa_man."

Sentence shuffling is performed by swapping the sentences between the texts in the same labels. The advantage of this method is that it does not need any additional resources, with low cost to implement. This also improves the results in some algorithms for the datasets (the third column of Table 3) for a machine learning-based approach compared to the baseline results. This method depends on the grammar structures of the texts (a disadvantage of free text in sentiment analysis problems) due to the effect on the performance of the sentence tokenizer. For deep neural network, sentence shuffling is only better in datasets 1 and 2 (Table 5 shows the results of deep neural network approach, the third column); this shows that the method is effective for less noise datasets and correct grammatical structures. The results of a Vietnamese EDA method are a little better than this method in the predictive models; the differences between them depend on the datasets and the applied algorithms.

The Google translator is considered to translate from Vietnamese to English in order to implement back translation techniques. This method is stable and the accuracy results are boosted among classifiers on the whole experimented datasets based on machine learning classifiers (the fourth column of Table 3). The results are better than the baseline result based on the deep learning model (the fourth column of Table 5). However, these results are over Vietnamese EDA and sentence shuffling for dataset 1; others are worse. The reason may be that dataset 1 contains fewer noise information than the remaining datasets. The noise information is harmful to the quality of the translated text and even has totally different meanings. Additionally, these results are better and more stable for almost all datasets when incorporating the mixup embedding (the fourth column of Table 6). This method generally depends on an effective machine translator; this approach obviously faces some challenges for sentiment analysis focusing on user-

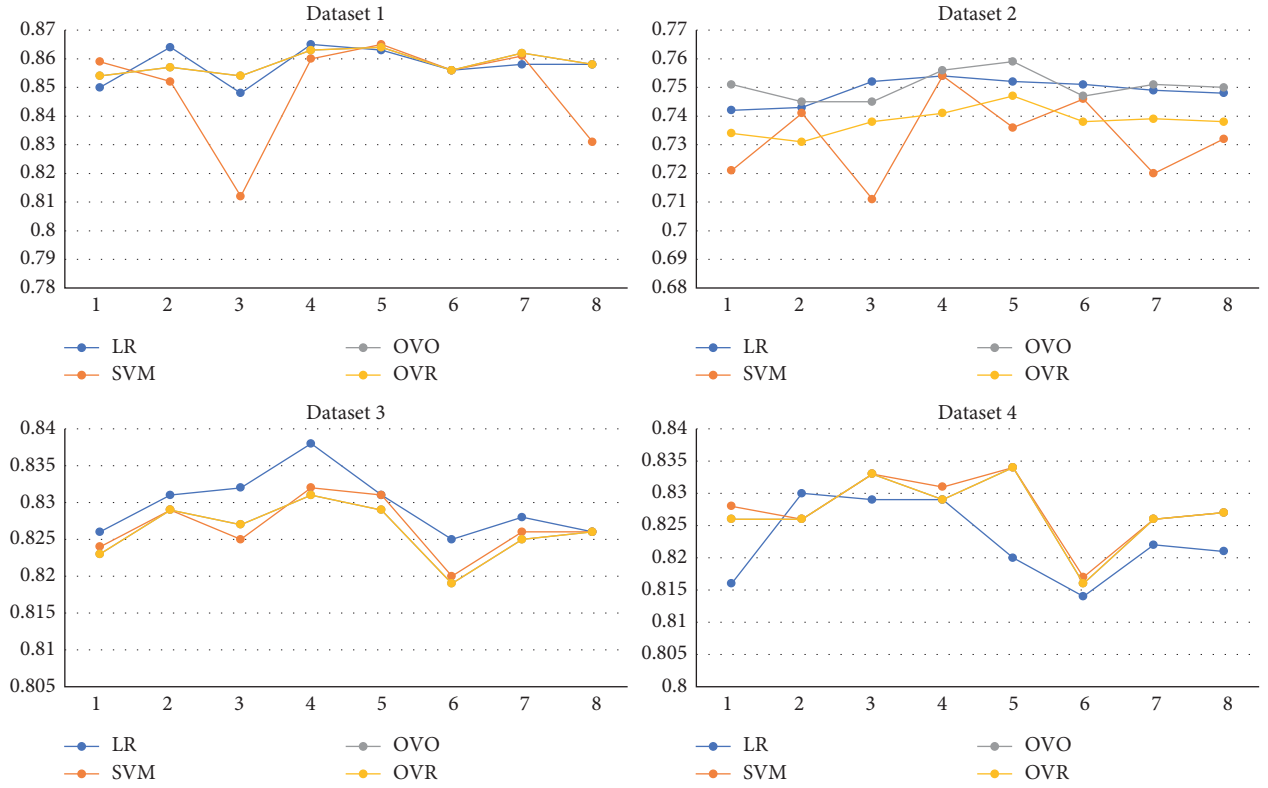

Figure 4: The accuracy results of various data augmentation techniques for Vietnamese sentiment analysis based on machine learning classifiers.

Table 5: The mean accuracy results (with standard deviation) of various data augmentation techniques for Vietnamese sentiment analysis in 10 runs based on the deep learning model.

Datasets	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Dataset 1	0.764 ± 0.05	0.838 ± 0.01	0.818 ± 0.03	0.843 ± 0.02	0.844 ± 0.01	0.840 ± 0.01	**0.847 ± 0.0003**
Dataset 2	0.415 ± 0.06	0.518 ± 0.03	0.438 ± 0.10	0.433 ± 0.07	**0.520 ± 0.01**	0.449 ± 0.05	0.367 ± 0.0003
Dataset 3	0.742 ± 0.05	0.797 ± 0.01	0.683 ± 0.10	0.742 ± 0.07	0.780 ± 0.01	0.782 ± 0.01	**0.801 ± 0.0001**
Dataset 4	0.757 ± 0.02	**0.802 ± 0.01**	0.710 ± 0.07	0.738 ± 0.09	0.782 ± 0.01	0.786 ± 0.01	0.746 ± 0.0001

(1) With preprocessing techniques; (2) EDA; (3) sentence shuffling; (4) back translation; (5) contextual substitution ($W2V$); (6) contextual substitution ($W2V + W2Vtf \times idf$); (7) sentence embedding mixup (the standard deviations are very minimum).

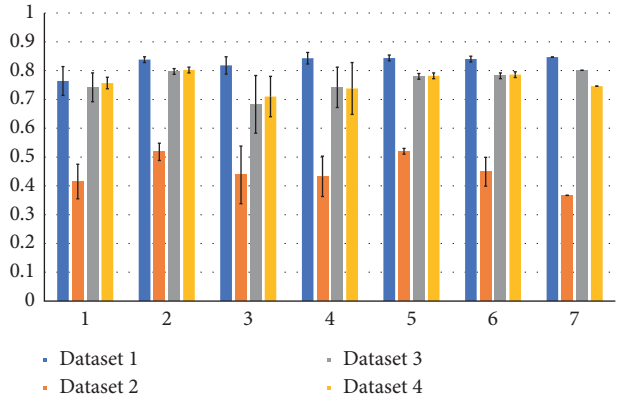

Figure 5: The mean accuracy results (with standard deviation) of various data augmentation techniques for Vietnamese sentiment analysis in 10 runs based on the deep learning model.

Table 6: The mean accuracy results (with standard deviation) incorporating the mixup method with other methods in 10 runs based on the deep learning model.

Datasets	(1)	(2)	(3)	(4)
Dataset 1	0.824 ± 0.0007	0.846 ± 0.0015	0.844 ± 0.0016	**0.854 ± 0.0002**
Dataset 2	**0.526 ± 0.0007**	0.389 ± 0.0011	0.396 ± 0.0003	0.466 ± 0.0003
Dataset 3	0.777 ± 0.0005	0.799 ± 0.0002	0.795 ± 0.0002	**0.820 ± 0.0002**
Dataset 4	0.796 ± 0.0002	0.786 ± 0.0007	**0.802 ± 0.0001**	0.774 ± 0.0003

(1) Mixup + EDA; (2) mixup + sentence shuffling; (3) mixup + $W2V$; (4) mixup + back translation.

generated text in digital communications without any constraints; it means that they contain a lot of wrong spellings, slang words, unknown words, and emoji icons and do not comply with the grammar structures.

The syntax-tree transformation also boosts the results steadily for all datasets (the fifth column of Table 3) compared with the baseline results, Vietnamese EDA, sentence shuffling, and back translation. However, it is also impacted by the quality of the texts and the cost of computation is expensive due to interacting with the syntax tree.

In order to leverage the abundance of Internet resources and especially the rapid development of the pretrained models, we evaluate the impacts of contextual substitution to reduce the dependence on the thesaurus for the low-resource languages like Vietnamese. For this, we have built pretrained models with more than 100 thousand of the collected reviews based on word2vec. An experiment will randomly choose n words for replacement and find their contextual words to replace them (the sixth column of Table 3 and the fifth column of Table 5). As a functional approach, some words might be replaced by the unsuitable words because they are near to each other on vector space, even opposite meaning. Thus, the next one assesses the impacts of choosing the replacing words in text. We choose the replacing words based on $tf \times idf$ scores which are higher; the information is less (the seventh column of Table 3 and the sixth column of Table 5).

For machine learning-based approach, $tf \times idf$-based substitution mostly improves better than random-based substitution and overcomes the baseline results (without augmenting data) (the first column of Table 3). For deep learning-based approach, the contextual substitutions promote the results steadily among datasets compared to the baseline results (the first column of Table 5) and the accuracy results of $tf \times idf$-based substitution are higher than random-based substitution in dataset 3 and dataset 4. These ones show that choosing the replacing words is an important clue and impacts the performances of the contextual substitution. In any cases, the $tf \times idf$-based substitution is lower; this may have some important words which have the low $tf \times idf$ scores because the frequencies of using them is dense among the training samples; for example, "hài_lòng" word (pleasant) is the important word which appears in most of the reviews in dataset 2 and it is also chosen for replacing word in dataset 2.

We also utilize MLM to obtain new samples based on PhoBERT. It also improves the results of the datasets (the eighth column of Table 3). This method needs sufficiently large data to build the pretrained model with BERT.

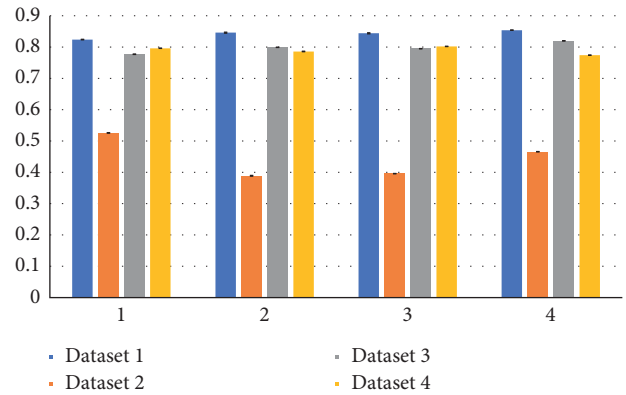

Figure 6: The mean accuracy results (with standard deviation) incorporating the mixup method with other methods in 10 runs based on the deep learning model.

The mixup approach is only performed on deep learning models by injecting as a mixup layer over the output layer. The seventh column of Table 5 shows the mixup results. This only promotes well for dataset 1 and dataset 3 and is a stable method having the very minimum standard deviation in 10 runs. In addition to applying the mixup layer, we incorporate this method with other augmented methods, including back translation, the contextual substitution, EDA, and sentence shuffling (Table 6 presents the results of these combinations) (see Figure 6). We achieve better results and they improve steadily among the datasets. These combinations obtain the best accuracy compared to other separable methods, including dataset 1 (the combination with back translation), dataset 2 (the combination with EDA), and dataset 3 (the combination with $W2V$).

5. Conclusions

The paper summarized the important methodologies, proposed some approaches in preprocessing techniques, performed many various experiments, and assessed the effectiveness of the approaches for Vietnamese sentiment analysis problem under low-resources training data. The predictive models are both some well-known machine learning-based classifiers and deep neural networks. The experiments execute the several preprocessing techniques to indicate the suitable techniques. The obtained results show that these preprocessing techniques are necessary; some techniques boost the performances significantly.

The limitation of training data is still one of the main barriers leading to the low performances of predictive

models. This paper has also presented the text augmentation techniques which enhance the text data from the original training data. These aim to obtain the generalization of predictive models and improve the results. As a result, text augmentation techniques achieve promising results. These help to generate more data and expand the knowledge of data, avoid overfitting, resolve unknown words, and lead to domain-independence. They are especially meaningful for the low-resource languages to develop the smart systems. These save time, money, and human resources to build the annotated datasets for the limited original training data. Many methods are simple to implement such as synonym replacement, insertion, deletion, and swapping randomly and have still achieved better performances. Some global approaches consist of back translation, the contextual substitution, and the embedding mixup which are the effective solutions to enhance the training data and can be applied for cross-languages.

In the future, we will investigate novel methods to augment quality data for sentiment analysis in particular and for text classification in general. In particular, we have been concerned with embedding space to leverage the pretrained models to obtain the synthetic samples instead of raw text.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank Ho Chi Minh City Open University. This work is a part of the authors' project under Grant E2019.02.3 funded by Ho Chi Minh City Open University.

References

[1] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: a survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, pp. 1942–4795, 2018, https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1253.

[2] P. Karuppusamy, "Analysis of neural network based language modeling," *Journal of Artificial Intelligence and Capsule*, vol. 2, no. 1, pp. 53–63, 2020, https://www.irojournals.com/aicn/V2/I1/06.pdf.

[3] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Systems with Applications*, vol. 77, pp. 236–246, 2017, https://linkinghub.elsevier.com/retrieve/pii/S0957417417300751.

[4] T. D. V. Vijayakumar, "Posed inverse problem rectification using novel deep convolutional neural network," *Journal of Innovative Image Processing*, vol. 2, no. 3, pp. 121–127, 2020, https://www.irojournals.com/iroiip/V2/I3/01.pdf.

[5] T. Zhou, T. Han, and E. L. Droguett, "Towards trustworthy machine fault diagnosis: a probabilistic Bayesian deep learning framework," *Reliability Engineering & System Safety*, vol. 224, 2022 https://linkinghub.elsevier.com/retrieve/pii/S095183202200179X, Article ID 108525.

[6] J. Caceres, D. Gonzalez, T. Zhou, and E. L. Droguett, "A probabilistic bayesian recurrent neural network for remaining useful life prognostics considering epistemic and aleatory uncertainties," *Structural Control and Health Monitoring*, vol. 28, no. 10, pp. 1545–2263, 2021, https://onlinelibrary.wiley.com/doi/10.1002/stc.2811.

[7] T. Singh and M. Kumari, "Role of text pre-processing in twitter sentiment analysis," *Procedia Computer Science*, vol. 89, pp. 549–554, 2016, https://linkinghub.elsevier.com/retrieve/pii/S1877050916311607.

[8] Z. Jianqiang and G. Xiaolin, "Comparison research on text pre-processing methods on twitter sentiment analysis," *IEEE Access*, vol. 5, pp. 2870–2879, 2017.

[9] S. Symeonidis, D. Effrosynidis, and A. Arampatzis, "A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis," *Expert Systems with Applications*, vol. 110, pp. 298–310, 2018, https://linkinghub.elsevier.com/retrieve/pii/S0957417418303683.

[10] K. V. Ghag and K. Shah, "Negation handling for sentiment classification," in *Proceedings of the 2016 International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1–6, Pune, India, August 2016.

[11] M. T. Al-Sharuee, F. Liu, and M. Pratama, "Sentiment analysis: an automatic contextual analysis and ensemble clustering approach and comparison," *Data & Knowledge Engineering*, vol. 115, pp. 194–213, 2018, https://linkinghub.elsevier.com/retrieve/pii/S0169023X17301866.

[12] U. Farooq, "Negation handling in sentiment analysis at sentence level," *Journal of Computers*, vol. 12, pp. 470–478, 2017, http://www.jcomputers.us/index.php?m=content&c=index&a=show&catid=187&id=2732.

[13] A. Hogenboom, D. Bal, F. Frasincar, M. Bal, F. De Jong, and U. Kaymak, "Exploiting emoticons in sentiment analysis," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13*http://dl.acm.org/citation.cfm?doid=2480362.2480498, Coimbra, Portuga, March 2013.

[14] H. Wang and J. A. Castanon, "Sentiment expression via emoticons on social media," *2015 IEEE International Conference on Big Data (Big Data)*, IEEE, in *Proceedings of the 2015 IEEE International Conference on Big Data (Big Data)*, pp. 2404–2408, November 2015, http://ieeexplore.ieee.org/document/7364034/.

[15] P. Kralj Novak, J. Smailović, B. Sluban, and I. Mozetič, "Sentiment of emojis," *PLoS One*, vol. 10, no. 12, https://doi.org/10.1371/journal.pone.0144296, Article ID e0144296, 2015.

[16] G. Guibon, M. Ochs, and P. Bellot, *From Emojis to Sentiment Analysis*, 2016.

[17] K. Kim, "An improved semi-supervised dimensionality reduction using feature weighting: application to sentiment analysis," *Expert Systems with Applications*, vol. 109, pp. 49–65, 2018, https://linkinghub.elsevier.com/retrieve/pii/S0957417418303166.

[18] K. Liu, D. Ergu, Y. Cai, B. Gong, and J. Sheng, "A new approach to process the unknown words in financial public opinion," *Procedia Computer Science*, vol. 162, pp. 523–531, 2019, https://linkinghub.elsevier.com/retrieve/pii/S1877050919320290.

[19] M. Abulaish and A. K. Sah, "A text data augmentation approach for improving the performance of CNN," in *Proceedings of the 2019 11th International Conference on Communication Systems & Networks (COMSNETS)*,

pp. 625–630, IEEE, Bangalore, India, January 2019, https://ieeexplore.ieee.org/document/8711054/.

[20] J. Wei and K. Zou, "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6382–6388, Hong Kong, China, November 2019.

[21] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the 54th Annual Meeting of the Association for Computational*, vol. 1, pp. 86–96http://aclweb.org/anthology/P16-1009, Berlin, Germany, August 2016.

[22] M. Fadaee, A. Bisazza, and C. Monz, "Data augmentation for low-resource neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 2, pp. 567–573http://aclweb.org/anthology/P17-2090, Vancouver, Canada, July 2017.

[23] A. Sugiyama and N. Yoshinaga, "Data augmentation using back-translation for context-aware neural machine translation," in *Proceedings of the Fourth Workshop on Discourse in Machine Translation*, pp. 35–44https://www.aclweb.org/anthology/D19-6504, Hong Kong, China, November 2019.

[24] M. Xia, X. Kong, A. Anastasopoulos, and G. Neubig, "Generalized data augmentation for low-resource translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5786–5796, Florence, Italy, July 2019.

[25] S. Kobayashi, "Contextual augmentation: data augmentation by words with paradigmatic relations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, pp. 452–457http://aclweb.org/anthology/N18-2072, New Orleans, Louisiana, June 2018.

[26] H. Guo, Y. Mao, and R. Zhang, "Augmenting Data with Mixup for Sentence Classification: An Empirical Study," 2019, https://arxiv.org/pdf/1905.08941.

[27] L. Sun, C. Xia, W. Yin, T. Liang, P. Yu, and L. He, "Mixup-transformer: dynamic data augmentation for NLP tasks," in *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 3436–3440https://www.aclweb.org/anthology/2020.coling-main.305, Barcelona, Spain, December 2020.

[28] B. Ding, L. Liu, L. Bing et al., "DAGA: data augmentation with a generation approach for low-resource tagging tasks," in *Proceedings of the 2020 Conference On Empirical Methods In Natural Language Processing (EMNLP)*https://www.aclweb.org/anthology/2020.emnlp-main.488, Punta Cana, Dominican Republic, November 2020.

[29] J. Wei, C.-Y. Huang, S. Xu, and S. Vosoughi, "Text augmentation in a multi-task view," *EACL*, vol. 7, 2021.

[30] K. T. L. Vnemolex, "A Vietnamese Emotion Lexicon for Sentiment Intensity Analysis," 2017, https://zenodo.org/record/801609.

[31] D. Quoc Nguyen and A. Tuan Nguyen, "PhoBERT: pre-trained language models for Vietnamese," *In Findings of the Association for Computational Linguistics: EMNLP*, pp. 1037–1042, 2020.

[32] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: beyond empirical risk minimization," in *Proceedings of the International Conference on Learning Representations*, Vancouver, BC, Canada, April 2018, https://openreview.net/forum?id=r1Ddp1-Rb.

[33] Huu-Thanh Duong, H.-T. Tram-Anh Duong, and T.-A. Nguyen-Thi, "A review: preprocessing techniques and data augmentation for sentiment analysis," *Computational Social Networks*, vol. 8, no. 1, pp. 2197–4314, 2021, https://computationalsocialnetworks.springeropen.com/articles/10.1186/s40649-020-00080-x.

[34] Huu-Thanh Duong, H.-T. Duong, and V. Truong Hoang, "A survey on the multiple classifier for new benchmark dataset of Vietnamese news classification," in *Proceedings of the 2019 11th International Conference on Knowledge and Smart Technology (KST)*, pp. 23–28, IEEE, Phuket, Thailand, January 2019, https://ieeexplore.ieee.org/document/8687509/.