# Linguistic-based Augmentation for Enhancing Vietnamese Sentiment Analysis

Cuong Nguyen Manh*, Hieu Pham Minh†, Hoang Do Van‡, Khanh Nguyen Quoc§,
Khanh Nguyen¶, Manh Tran Van‖ and Anh Phan**
Faculty of Information Technology, Le Quy Don Technical University
Hanoi, Vietnam
Email: *cuongtop45@gmail.com, †emcuachi99@gmail.com, ‡boykutehyvn1999@gmail.com, §quockhanhktqs@gmail.com,
¶tungkhanhmta@gmail.com, ‖tranvanmanh2792@gmail.com, **anhpv@mta.edu.vn

***Abstract*— Identify customer's opinions about products, services, and brands bring many benefits to e-commerce development. Capturing customer attitudes helps retailers adjust business decisions. Customers can select the suitable product and the good service by consulting social experiences. However, free-style texts of customer feedback like acronyms, slang words, incorrect grammar, and so on are challenging any machine learning model.**

**This paper aims to generate the informative input text to boost the Vietnamese sentiment classifiers to e-commerce. Based on the data which observe on several e-commerce websites we propose preprocessing techniques and a data augmentation method. The experimental results on the data collected from popular e-commerce websites including Tiki, Shopee, and Lazada have shown that our methods can improve the performance of all classifiers significantly. We also provide the corpus to motivate the further research.**

## I. Introduction

Surveying customers' opinions may contribute significantly to e-commerce development. Recently, e-commerce is booming globally, especially, during the coronavirus pandemic, the demands for shopping online have continuously increased. Although, a main disadvantage of e-commerce is the difficulty to select the expected item from numerous ones provided by different services [4]. To make buying decisions, a customer normally bases on social knowledge revealed by their comments when experiencing products. The services also can analyze customer feedback to adjust business strategies. For such reasons, sentiment analysis in e-commerce has received the special attention of both researchers and practitioners.

Machine learning has been applied successfully to sentiment analysis problems. The traditional learning models are built on handcrafted features such as TF-IDF (term frequency-inverse document frequency), count vectors, bag of words (BoW), and so on [1]. Recently, deep neural networks have achieved the top performance in various domains by the ability to learn sophisticated features with little or without expert knowledge. The popular models for natural language processing are based on RNN (recurrent neural networks) architectures such as LSTM (long short term memory) and GRU (gated recurrent unit), and transformers [15]. For machine learning, the input data quality is one of critical criteria to achieve a good result.

Processing text data on e-commerce websites is challenging for any machine learning model, even for humans. Firstly, as common issues for social websites, customers may insert many special characters into their feedback like emotional symbols, URLs, hashtags. Secondly, since the user's text comments in the communication style, teen slang words, abbreviations, and non-diacritical marks, mixed languages (Vietnamese, English) are used. Additionally, the comments are made by diverse users with different purposes, the text grammar may be incorrect or irregular as the formal language. For example, a comment "vua re vua hin. toi gi k mua" contains no diacritical marks, a slang word "hin" − "xịn", an abbreviation of "không" − k. The comment can be meant as "vừa rẻ vừa xịn. tội gì không mua" (both cheap and good, should buy). For these issues, extracting semantic features from raw texts on e-commerce data is a really difficult task. This hinders the applications of both traditional learning and deep learning as well.

To solve problem mention above, this paper proposes a new pre-processing technique and an augmentation method to facilitate semantic features relevant to the sentiment in texts. Since the better input data quality, the classification performance is boosted significantly for both traditional learning algorithms and deep learning. The main contributions are summarized as follows:

- Introducing a new Vietnamese corpus for sentiment analysis for e-commerce[1].
- Proposing data preprocessing methods for social text to refine input data for learning models.
- Proposing a data augmentation technique to process texts without diacritical marks.

The rest of this paper is organized as follows: Section II surveys studies related to Vietnamese sentiment analysis problems. The proposed methods are described in Section III. Section IV presents the dataset, evaluation measures, experiments, and result discussion. Section V concludes our work.

## II. Related works

Sentiment analysis of the raw text is a hot topic in natural language processing and has been studied for a long time

978-1-6654-0435-8/21/$31.00 ©2021 IEEE

[1]https://github.com/cuongtop4598/Linguistic-based-Augmentation-Dataset

[12]. Many studies have shown the state of the art results for sentiment analysis in English. However, there are not many studies for the Vietnamese corpus. This section summarizes studies that are related to Vietnamese sentiment analysis.

In [8], Kieu, et. al introduced a rule-based system for Vietnamese sentiment analysis. The author's idea is matching words in the sentence in the positive word and negative word dictionary. After that, some rules were used to classify the sentiment of the sentence. The disadvantage of this approach is heavily depended on the diversity of the dictionary and the strength of the rules. Bach, et. al proposed a model to identify comparative sentences and recognize relations in them. Such information was used to analyze the sentiment of a sentence [2]. In [13], Trinh, et. al proposed a lexicon-based method for sentiment analysis with Facebook data. A Vietnamese emotional dictionary was built and a support vector machine is used to learn the sentiment of the sentences. This approach used handcraft features extracted from the text so it is hard to capture the high-level feature in the sentence. To automatically learn such features of input sentences, Vo, et. al used CNN and LSTM to generate information channels for Vietnamese sentiment analysis [14]. In this approach, CNN and LSTM are combined together for integrating the advantages. CNN was used to capture local dependencies between neighbor words by using convolutional filters, LSTM was used to preserve information over a long period of time. In [10], Nguyen, at. al applied hierarchical attention networks for Vietnamese document classification. The network has a two-level architecture with attention mechanisms applied to the word level and sentence level from which it reflects the hierarchical structure of the document.

In this work, we only used LSTM and proposed a preprocessing method which helps to improve the performance of the LSTM. In particular, we proposed a method to overcome the long term dependencies in LSTM by emphasizing the emotional adjectives in the sentence. The details of the method are presented in Section III.

A practical problem is Vietnamese users often type their comments without using diacritical marks to save time, e.g."Tôi hài lòng với sản phẩm này" becomes "Toi hai long voi san pham nay". Sometimes some words in the sentence are missing diacritical marks due to customer mistakes. These things make it difficult for the sentiment analysis process. This paper also presents a data augmentation method that helps to tackle these problems.

## III. Proposed methods

This section presents our proposed methods to facilitate text-based models for Vietnamese sentiment analysis. Our methods include (1) preprocessing techniques to overcome the long-term dependencies problem in RNNs (Section III-A), and (2) data augmentation to enrich information in the case of missing diacritical marks of words (Section III-B). The network architecture is described in section III-C.

### A. Preprocessing

*1) Long term dependency problem in Vietnamese sentiment analysis:* RNN-based networks are commonly applied to natural language processing tasks. An RNN network is designed with loops in them, allowing information to persist, i.e. allowing the previous inputs to affect the subsequent predictions. The architecture of the basic RNN is shown in Fig. 1, where $x_i$ and $h_i$ is the input and output of a cell $A$ at time $t_i$ $(i = \overline{0,t})$, respectively. These architecture guarantees that the output $h_j$ at time $t_j$ is affected by all the input $x_k$ where $(0 \leq k < j)$.

The main drawback of this architecture is the long-term dependencies problem. The problem occurs in case the output $h_j$ at time $t_j$ strongly depends on an input $x_k$ at time $t_k$but the distance from $t_j$ and $t_k$ is quite large so the information at $t_k$ is forgotten. The problem was explored in depth in [3].

In sentiment analysis, the sentiment of a sentence is the output $h_t$ of the last cell in the word sequence. The $h_t$ is called a sentence encoding vector which is expected to capture all the information of the entire sentence. In Vietnamese, to precisely predict the sentiment of a sentence, the $h_t$ should focus on some adjectives in the sentence which express the emotion of the customers. These words can appear at the beginning or in the middle of the sentence (e.g. "Tôi thích sản phẩm này!" ("I like this item"), the adjective "thích" ("like") is in the middle of the sentence) so the information of this words may be forgotten at the end of the sentence, i.e. $h_t$ forgets information of the crucial adjective in the sentence. To tackle this problem, we propose a preprocessing method which is presented in the next section.

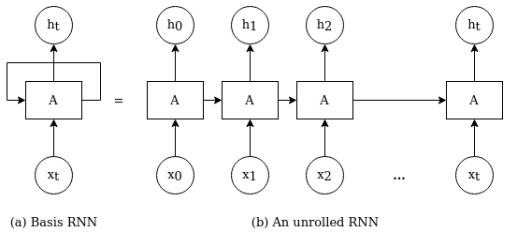


Figure 1. RNN architecture.

*2) Proposed preprocessing:* To prevent $h_t$q forgets some crucial adjectives in the sentence, we add an emotion word, *"positive"* or *"negative"*, at the end of the sentence based on the emotion expressed by the adjective in the sentence. Intuitively, these addition reminds the $h_t$ to remember some important adjective in the sentence.

To determine the expressed emotion by the adjectives in a sentence, we first create five dictionaries: PosWord which contains positive words in Vietnamese, PosIcon which contains positive icons, NegWord which contains negative words in Vietnamese, NegIcon which contains negative icons, and NotWord which contains negation words in Vietnamese. After that, based on the characteristics of the Vietnamese language, we create the rules to specify the emotion of the sentence $S$ as follows:

- $S$ containing words which are only in PosWord dictionary $\rightarrow$ Emotion is Positive
- $S$ containing words which are only in PosIcon dictionary $\rightarrow$ Emotion is Positive
- $S$ containing words which are only in NegWord dictionary $\rightarrow$ Emotion is Negative
- $S$ containing words which are only in NegIcon dictionary $\rightarrow$ Emotion is Negative
- $S$ containing the pattern: a word in NegWord + a word in PosWord $\rightarrow$ Emotion is Negative
- $S$ containing the pattern: a word in NegWord + a word in NegWord $\rightarrow$ Emotion is Positive

We add the word *"positive"* or *"negative"* at the end of the sentence if the emotion of the sentence is positive or negative, respectively. Fig. 2 illustrates our preprocessing method. Suppose the word $x_2$ expresses the emotion of the sentence. Based on this emotion, the sentence is expanded by adding an emotion word *"positive"* or *"negative"* at the end of the sentence.

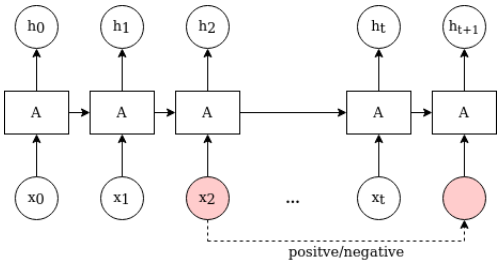


Figure 2. RNN with preprocessing.

### B. Data augmentation

By carefully looking at the data crawled from some Vietnamese e-commerce websites, we realize that some sentences in the dataset are removed diacritical marks. This may be due to the customers intentionally remove diacritical marks to save their time or possibly due to their typing errors. It becomes a challenge to predict the sentiment of these sentences. It also becomes a big obstacle for building a product evaluation system for e-commerce websites based on sentiment analysis because in real life missing diacritical marks often occur.

To overcome this challenge, we propose a data augmentation method that helps to analyze the sentiment of a sentence even in case it is misspelled or be removed diacritical marks. To do this, from the original dataset, we tried to randomly remove the diacritical mark of some words which simulates the customer's failure when typing their comment. This simulation is described in Algorithm 1. The algorithm takes original dataset $S$ as the input and outputs a new dataset $S'$. Each sentence $s_i$ in $S$ can create a new sentence $new\_sent$. After that, both sentences $s_i$ and $new\_sent$ is added to the $S'$ (line 23). Line 5 to line 9 is the random selection of $n\_rand$ positions in the sentence. Line 10 to 17 presents the process of creating $new\_sent$. The idea behind is to remove diacritical marks of the words at chosen positions in the sentence $s_i$.

The proposed data augmentation method help to capture more failure cases where sentence lack diacritical marks.

**Algorithm 1** Data augmentation

**INPUT**: Sentence dataset $S$ = $s_1$, $s_2$ ... $s_N$
**OUTPUT**: New sentence dataset $S'$

1: Initialize new sentence dataset: $S' = \{\emptyset\}$
2: **for** $s_i$ in $S$ **do**
3: $new\_sent = \{\emptyset\}$
4: $rand\_pos = \{\emptyset\}$
5: $n\_rand \leftarrow$ random number in [1, len($s_i$)]
6: **for** $k$ in range $n\_rand$ **do**
7: $pos \leftarrow$ random number in [1, len($s_i$)]
8: $rand\_pos$ = $rand\_pos \cup pos$
9: **end for**
10: **for** $word_j$ in $s_i$ **do**
11: **if** $j$ in $rand\_pos$ **then**
12: $new\_word \leftarrow$ remove diacritical mark of $word_j$
13: $new\_sent = new\_sent \cup new\_word$
14: **else**
15: $new\_sent = new\_sent \cup word_j$
16: **end if**
17: **end for**
18: $S' = S' \cup s_i \cup new\_sent$
19: **end for**

### C. Network architecture

Fig. 3 shows the network architecture used in this work. To classify the sentiment of the sentence, we stack a $Dense$ layer contains 2 neural with softmax activation function on the top of $h_t$. This function evaluates the probability $a_i$ (for $i = 0, 1$) as follows, where $a_0$ and $a_1$ are the probability of being positive and negative, respectively. If $a_0 \geq 0.5$, we conclude positive and vice versa.

$$a_i = \frac{exp(o_i)}{exp(o_0) + exp(o_1)} \quad \text{for } i = 0, 1 \tag{1}$$

where $o_1$, $o_2$ are the two values of the $Dense$ layer.

In RNN, a cell is a primary block that is repeated through time and denoted as $A$ in Fig. 3. There are some types of RNN cell. In this paper, we use two popular types of them, i.e Long Short Term Memory (LSTM) is introduced by Hochreiter, et al. in [7], and Gated Recurrent Unit (GRU) is introduced by Cho, et al. in [5].

These mentioned RNN architectures allow learning only through one direction of the time, i.e from $x_1$ to $x_t$. This is a drawback of RNN because it can not learn the information in the opposite direction. To improve the performance of RNN, an extension of RNN is bidirectional RNN [11] was proposed. The network can be trained simultaneously in the positive and negative time direction. In this paper, we also use BiLSTM and BiGRU to carry out some experiments. The comparison of the performance of these network architectures is presented in the next section.

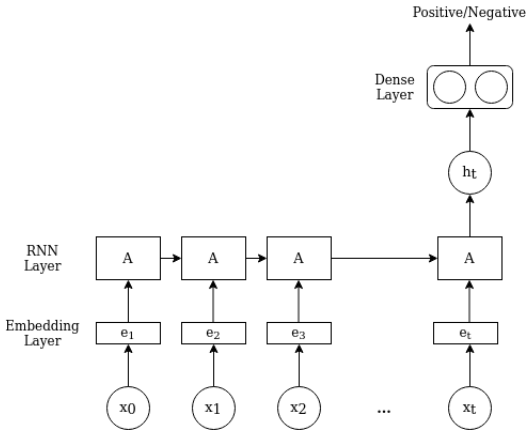


Figure 3. RNN for Vietnamese sentiment analysis.

Table I
DATASET

| | Positive | Negative |
|---|---|---|
| # training | 8000 | 8000 |
| # testing | 2000 | 2000 |

## IV. EXPERIMENTS

### A. Dataset

The dataset conduct experiments is collected from three popular e-commerce websites in Vietnam including Tiki[2], Shopee[3], and Lazada[4]. The collection was accomplished by using Selenium[5] which contains tools and libraries that enable and support the automation of web browsers. Each sentence was annotated by four persons and checked the agreement. We then used some simple preprocessing such as removing stopwords and word segmentation[6].

The dataset has a total of twenty thousand feedback sentences from customers. Each sentence can be positive or negative which is labeled by hand. The longest sentence contains 200 words. The dataset is randomly divided into training and test sets. Table I shows the data statistics.

### B. Evaluation measures

The performance of the classifiers was measured using the Accuracy and the F1 score that are suitable in the case of balanced data.

$$accuracy = \frac{\#\text{ of correctly classified sentences}}{\#\text{ of sentences}} \quad (2)$$

The F1 score is calculated based on precision and recall. The F1 score was evaluated on each type of sentence. Let take sentences belonging to the positive as an example, precision, recall, and the F1 score was measured as follows:

$$precision = \frac{\#\text{ of correctly classified positive sentences}}{\#\text{ of predicted positive sentences}} \quad (3)$$

$$recall = \frac{\#\text{ of correctly classified positive sentences}}{\#\text{ of positive sentences}} \quad (4)$$

$$F1 = \frac{2 * precision * recall}{precision + recall} \quad (5)$$

### C. Experimental settings

We used Keras framework [6] for our implementation.

32,000 sentences were randomly generated by applying proposed data augmentation, four times the original dataset. The number of words in our five dictionaries is shown in Fig. 4.

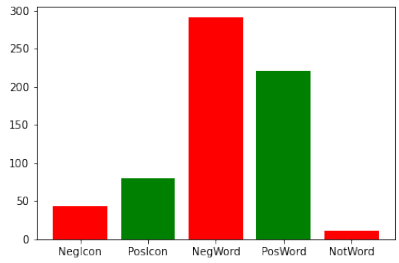


Figure 4. Statistic on the dictionaries

The RNN-based networks are depicted in Fig. 3. We set the input sequence length to 200 because the longest sentence contains 200 words. For shorter sentences, zero paddings are used. The size of each embedding vector $e_i$ is set to 200.

In the training phase, the Adam optimization algorithm (with learning rate = 0.001 and batch size = 32) is used to learn model parameters. Details of Adam optimization can be found in [9].

### D. Experimental results

Tables II and III present the classification performance when applying our proposed methods. ♦ and ▲ denote the applications of the preprocessing techniques and data augmentation; − denotes the data with diacritical marks removed. We ran each model 5 times and took the average.

In Table II, each classifiers reaches the similar performance two cases of original texts and texts with diacritical marks removed. As aforementioned, the corpus made by diverse users, so that the sentences may contain many incorrect words related to typos, abbreviations, slang, without diacritics. These issues challenge both traditional learning and deep learning models. For traditional learning, statistical feature extractors generate the wrong values, since they can not capture the changed words. For deep learning networks, e.g. LSTM,

[2]https://tiki.vn/
[3]https://shopee.vn/
[4]https://www.lazada.com
[5]https://www.selenium.dev/documentation/en/webdriver/
[6]Underthesea, a Vietnamese NLP Toolkit - https://github.com/undertheseanlp/underthesea

Table II
PERFORMANCE COMPARISON IN THE CASE OF APPLYING THE PREPROCESSING TECHNIQUE

| | Accuracy$^{-}$ | Accuracy | Accuracy$^{\blacklozenge}$ | F1 | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Positive$^{-}$ | Positive | Positive$^{\blacklozenge}$ | Negative$^{-}$ | Negative | Negative$^{\blacklozenge}$ |
| DT | 79.6 | 80.5 | **85.12**(+4.62) | 80.0 | 80.8 | **85.4**(+4.6) | 79.1 | 80.1 | **84.8**(+4.7) |
| NB | 68.6 | 71.9 | **78.3**(+6.4) | 56.0 | 63.1 | **74.0**(+10.9) | 75.0 | 77.2 | **81.4**(+4.2) |
| RF | 87.1 | 86.3 | **89.8**(+3.5) | 86.9 | 86.1 | **89.9**(+3.8) | 87.2 | 86.4 | **89.7**(+3.3) |
| LSTM | 88.9 | 90.2 | 90.1(−0.1) | 89 | 90.2 | 91.1(+0.9) | 88.2 | 90.2 | 91.16(+0.96) |
| GRU | 88.5 | 88.8 | **91.5**(+2.7) | 88.4 | 88.7 | **91.63**(+2.93) | 88.8 | 88.9 | **91.3**(+2.4) |
| BiLSTM | 88.2 | 89.3 | **91.4**(+2.1) | 88.1 | 89.2 | **91.63**(+2.43) | 88.4 | 89.3 | **91.31**(+2.01) |
| BiGRU | 88.3 | 88.9 | **91.29**(+2.39) | 88.4 | 88.7 | **91.28**(+2.58) | 88.2 | 89.2 | **91.3**(+2.1) |

Table III
THE PERFORMANCE COMPARISON IN THE CASE OF APPLYING THE PREPROCESSING AND DATA AUGMENTATION

| | Accuracy$^{\blacktriangle}$ | Accuracy$^{\blacktriangle\blacklozenge}$ | F1 | | F1 | |
|---|---|---|---|---|---|---|
| | | | Positive$^{\blacktriangle}$ | Positive$^{\blacktriangle\blacklozenge}$ | Negative$^{\blacktriangle}$ | Negative$^{\blacktriangle\blacklozenge}$ |
| DT | 79.2 | **85.27**(+6.07) | 79.1 | **85.4**(+6.3) | 79.3 | **85.05**(+5.75) |
| NB | 68.6 | **72.8**(+4.2) | 57.0 | **64.9**(+7.9) | 75.3 | **77.7**(+2.4) |
| RF | 83.4 | **89.8**(+6.4) | 83.0 | **89.9**(+6.9) | 83.7 | **89.7**(+6.0) |
| LSTM | 88.8 | 88.9(+0.1) | 88.7 | 88.8(+0.1) | 88.8 | 88.9(+0.1) |
| GRU | 89 | 89.2(+0.2) | 89.05 | 89.2(+0.15) | 89.07 | 89.2(+0.13) |
| BiLSTM | 89 | 88.6(−0.4) | 89.2 | 88.8(−0.4) | 88.7 | 88.4(−0.3) |
| BiGRU | 89 | 89.9(+0.9) | 89.1 | 89.8(+0.7) | 88.9 | 90.1(+1.2) |

changed words lead to the loss of sequential connections, thus degrading classification performance.

Our text processing methods boost classifiers significantly. Noticeably, the Naive Bayes classifier is improved 6.4% in Accuracy and 10.9% in F1 of the positive; the decision tree is improved 4.62% in Accuracy and 4.6% in F1. Predicting sentimental adjectives and emotions, and emphasizing them at the end of sentences facilitate learning models. Given two examples 1) `Giao hàng quá` **`chậm`** `hôm nào cũng phải vào hóng xem hàng của mình đi đến đâu rồi`, and 2) `Tôi` **`không hài lòng`** `một chút nào về shop, mất niềm tin vào mua hàng online`, we detect the sentimental adjectives (the bold words) and add them to the end. For traditional learning, such words are put more weight. For sequence-based networks, it avoids the words being forgotten by long dependencies. As a result, the classifiers achieve better performance.

To enrich the data with the aim to predict the sentiment for sentences with or without diacritics, we propose a data augmentation as presented in Algorithm 1. We create new sentences by randomly removing diacritical marks, and add to the dataset. Table III compares classification performance when training models on the augmented data with the preprocessing technique. Compatible with the results in Table II, emphasizing sentimental adjectives is beneficial to the learning process. After analyzing the prediction, we have found that many samples with diacritics missed are predicted incorrectly by the models with the original dataset, but are predicted correctly by the models trained on augmented data. Some examples are as follows:

`Troi oi! Chức năng này mà minh rat thích tren iPhone`

`Cuc ky hai long sau gan 10 ngay su dung Hang mới tinh, con nguyen seal, mo ra check thì moi bao kich hoat bao hanh 2-3 ngay, 100% xin, moi, zin.`

## V. CONCLUSION

In this paper, we present preprocessing techniques and a data augmentation method. These proposals make machine learning models can train on defective texts. Emphasizing sentimental words facilitates the feature extractors in traditional learning and addresses the long dependencies in sequence-based models. The experimental results have proved the efficiency of our proposals and potential application for Vietnamese texts on social websites.


## ACKNOWLEDGMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2018.306


## REFERENCES


[1] Ravinder Ahuja, Aakarsha Chug, Shruti Kohli, Shaurya Gupta, and Pratyush Ahuja. The impact of features extraction on the sentiment analysis. *Procedia Computer Science*, 152:341–348, 2019.

[2] Ngo Xuan Bach, Pham Duc Van, Nguyen Dinh Tai, and Tu Minh Phuong. Mining vietnamese comparative sentences for sentiment analysis. In *2015 Seventh international conference on knowledge and systems engineering (KSE)*, pages 162–167. IEEE, 2015.

[3] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[4] Xusen Cheng, Linlin Su, and Alex Zarifis. Designing a talents training model for cross-border e-commerce: a mixed approach of problem-based learning with social media. *Electronic Commerce Research*, 19(4):801–822, 2019.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] François Chollet et al. Keras. https://keras.io, 2015.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[8] Binh Thanh Kieu and Son Bao Pham. Sentiment analysis for vietnamese. In *2010 Second International Conference on Knowledge and Systems Engineering*, pages 152–157. IEEE, 2010.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Khanh Duy Tung Nguyen, Anh Phan Viet, and Tuan Hao Hoang. Vietnamese document classification using hierarchical attention networks. In *Frontiers in Intelligent Computing: Theory and Applications*, pages 120–130. Springer, 2020.

[11] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[12] Mohammad Soleymani, David Garcia, Brendan Jou, Björn Schuller, Shih-Fu Chang, and Maja Pantic. A survey of multimodal sentiment analysis. *Image and Vision Computing*, 65:3–14, 2017.

[13] Son Trinh, Luu Nguyen, Minh Vo, and Phuc Do. Lexicon-based sentiment analysis of facebook comments in vietnamese language. In *Recent developments in intelligent information and database systems*, pages 263–276. Springer, 2016.

[14] Quan-Hoang Vo, Huy-Tien Nguyen, Bac Le, and Minh-Le Nguyen. Multi-channel lstm-cnn model for vietnamese sentiment analysis. In *2017 9th international conference on knowledge and systems engineering (KSE)*, pages 24–29. IEEE, 2017.

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.