DEEP LEARNING MODEL WITH HIERARCHICAL ATTENTION MECHANISM FOR SENTIMENT CLASSIFICATION OF VIETNAMESE COMMENTS

Luu Van Huy
The University of Danang,
University of Science and Technology,
Viet Nam
lvhuy@dut.udn.vn

Ngo Le Huy Hien
Leeds Beckett University,
United Kingdom
ngolehuyhien@gmail.com

Nguyen Thi Hoang Phuong
Pham Van Dong University
Viet Nam
nthphuong@pdu.edu.vn

Nguyen Van Hieu*
The University of Danang,
University of Science and Technology
Viet Nam
nvhieuqt@dut.udn.vn

Article history:
Received 03.04.2023, Accepted 15.09.2023

Abstract

In the current digital era, text documents become valuable for businesses to reach potential customers and curtail advertising costs. However, extracting and classifying beneficial information from texts can prove challenging and time-consuming, particularly in complex languages like Vietnamese. This study aims to classify the sentiment of Vietnamese comments on e-commerce websites into negative and positive classes. To enhance the performance of sentiment classification, the study fine-tuned traditional models of Convolutional Neural Networks and Recurrent Neural Networks (RNN). Then, this research proposed a combination of RNN and attention mechanisms at the word and word-and-sentence levels of the input document. The results showed an impressive accuracy of 93.72% and an F1 score of 93.7% on the RNN model with a word-and-sentence-level attention mechanism. This research outcome contributes to the field of text classification and could be applied in opinion mining, customer feedback analysis, and natural language processing. Future work aims to enhance sentiment analysis accuracy and expand the models' scope to encompass more languages.

Key words

Sentiment Classification, Convolutional Neural Network, Deep Learning Models, Hierarchical Attention Mechanism, Word-level, Word-and-sentence-level, Vietnamese Comments.

1 Introduction

Nowadays, with the proliferation of electronic information and e-commerce platforms such as social networks like Facebook, Twitter, and e-commerce sites like Amazon, and Alibaba, the Internet is not only a source of information and a place to buy goods, but also a medium where users express their emotions, exchange experiences, and try out services [Vu et al., 2018]. Understanding users' emotions expressed in conversations help businesses analyze their customers' level of interest in their brands, products, and services. Therefore, data sources with emotional tones are widely analyzed in recent years, especially in social media, product review blogs, websites, and call center chat channels [Hien et al., 2020]. User emotions are commonly divided into categories such as positive and negative or classified by levels of happiness, neutrality, and anger.

The traditional data exploration field primarily focuses on analyzing a user's history, such as purchase history or access time, while the user emotion exploration field focuses on analyzing the meaning of comments on information pages or social networks [Jain et al., 2020]. Therefore, classifying user emotions is a multidisciplinary problem that requires combining data exploration and natural language processing techniques [Shats et al., 2022]. Despite the various approaches and levels of processing developed for sentiment classification, the majority of existing research has been focused on the English language, leaving many limitations for processing Vietnamese documents.

To address this gap, this study aims to classify the sentiment of user comments on e-commerce websites into 2 main categories, positive and negative sentiment. Various Deep Learning models in text classification, which achieved impressive results, are examined and experimented with in this study. Furthermore, the proposed model has the potential of attention mechanisms when applied to natural language processing tasks, particularly sentiment classification of comments in a high-complexity language like Vietnamese. This research contributes to the development of the field of sentiment recognition and serves the classification and retrieval needs of individuals or businesses.

2 Related Works

The sentiment classification of online comments has been the subject of numerous studies in recent years, with increasingly large datasets collected from e-commerce websites across the globe. One approach to sentiment classification, known as the dictionary-based method, relies on sentiment-related words to predict emotions in text [Ameur et al., 2005]. This method involves identifying individual sentiment words, assigning scores to positive and negative words, and summing up these scores based on a defined metric to determine the overall emotional tone of the text. This approach is straightforward to implement and computationally cost-efficient, requiring only the effort of building the sentiment-word dictionary. However, one major limitation of the approach is the omission of word order, which can result in important information being lost. Additionally, the accuracy of the model is dependent on the quality of the built-in sentiment word dictionary.

Another approach to sentiment classification combines rule-based and corpus-based methods [Im et al., 2015], as demonstrated by Richard Socher and his colleagues at Stanford University [Socher et al., 2013]. They utilized a Deep Learning Recursive Neural Network (DL-RNN) in conjunction with a natural language processing knowledge base referred to as the Sentiment Treebank [Socher et al., 2013]. The Sentiment Treebank is a syntactic parse tree of a sentence, in which each node in the tree is associated with a sentiment weight set, indicating the sentiment of the subtree rooted at that node. The weight is classified on a five-point scale, from very negative, negative, neutral, positive, to very positive [Habib et al., 2022]. The label with the highest weight determines the overall label of the node. While the combination of the DL-RNN and Sentiment Treebank provides a more sophisticated approach to sentiment classification, it also requires greater computational resources and knowledge of natural language processing. Nevertheless, it offers the advantage of a more nuanced representation of sentiment in text, incorporating both semantic and syntactic information.

With the advancements in CPU and GPU processing speed and the reduction in hardware costs [Bazhanov et al., 2018], cloud computing infrastructure services have experienced a proliferation, providing opportunities for the development of Deep Learning Neural Networks (DLNNs) as a learning method [Hien et al., 2022]. The advantage of this method lies in its capability to predict inputs as one sentence or one paragraph. Deep Learning models have garnered noteworthy achievements in computer vision [Hinton et al., 2012] [Graves et al., 2013]. Convolutional neural network models originally developed for computer vision were later proven to be effective for natural language processing (NLP) and achieved outstanding outcomes in semantic analysis [Yih et al., 2011], query retrieval [Shen et al., 2014], sentence modeling [Kalchbrenner et al., 2014], and other traditional natural language processing tasks [Collobert et al., 2011]. In the process of natural language processing, the majority of work uses deep learning models like CNNs to learn vector representations through neural language models [Bengio et al., 2003][Yih et al., 2011] [Mikolov et al., 2013] and process the learned vectors for classification [Collobert et al., 2011]. Therefore, this study proposes two models inspired by prior research that used Convolutional Neural Networks for text classification [Zhang et al., 2019][Liu, 2020].

Despite the remarkable achievements in the field of computer vision, Deep Neural Networks can only be applied to problems where inputs and targets can be reasonably encoded by fixed-dimensional vectors. This is a significant limitation in natural language processing problems, where data is represented by sequences whose length is unknown beforehand. Some studies have employed Recurrent Neural Networks (RNN) [Hu et al., 2020] and its popular variants such as Long Short-Term Memory (LSTM) [Hochreiter et al., 1997] and Gated Recurrent Unit (GRU) [Cho et al., 2014] architectures to address common issues related to sequence-to-sequence.

Alongside the advancements in neural networks aiming to optimize text classification tasks, another approach is the attention mechanism, which has been widely applied in all types of Natural Language Processing (NLP) tasks based on Deep Learning [Hien et al., 2021]. The essence of the attention mechanism is that it mimics the human visual attention mechanism. When the human visual attention mechanism detects an object, it typically does not scan the entire scene from beginning to end; instead, it focuses on a particular part as required by the person. Initially, the attention mechanism was primarily applied in the field of visual recognition in the 1990s. However, it did not become a trend until Google DeepMind applied the attention mechanism to Recurrent Neural Network (RNN) models for image classification [Mnih et al., 2014]. Then, many researchers experimented with the attention mechanism for machine translation tasks. Notably, Dzmitry Bahdanau and his colleagues applied this approach for simultaneous translation [Bahdanau et al., 2014]. They were the first to apply the attention mechanism to NLP. Their research received recognition and the attention mechanism subsequently became prevalent in NLP tasks based on neural networks such as RNN or Convolutional Neural Networks (CNN).

In this paper, the research aims to optimize and combine Deep Recurrent Neural Network and attention mechanism, with the expectation of focusing on the important and emotional-emphasized aspects of the text, improving the overall performance of sentiment classification.

3 Proposed Model and Methodology

3.1 Convolutional Neural Network model with built-in Residual classifier of Deep Learning Neural Network

Firstly, this study builds a model designed for sentiment classification of comments, which employs Convolutional Neural Networks (CNNs) (depicted in Figure 1). Specifically, the model is built upon a previously established neural network model for text classification [Zhang et al., 2019], supplemented with Residual Networks layers (as presented in Figure 2). By utilizing Residual Networks layers [He et al., 2016], the neural network architecture can be constructed with greater depth, thereby enabling a more manageable differentiation process. Consequently, it is anticipated that the model will yield a notable improvement in the accuracy of sentiment classification of comments.

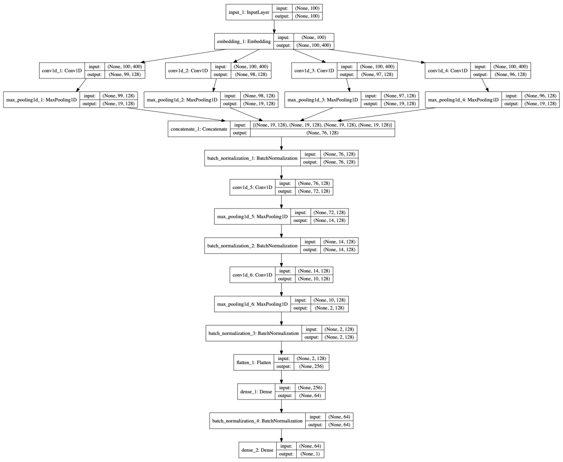

input_1: InputLayer | input: (None, 100) | output: (None, 100)
embedding_1: Embedding | input: (None, 100) | output: (None, 100, 400)
conv1d_1: Conv1D | input: (None, 100, 400) | output: (None, 99, 128)
conv1d_2: Conv1D | input: (None, 100, 400) | output: (None, 98, 128)
conv1d_3: Conv1D | input: (None, 100, 400) | output: (None, 97, 128)
conv1d_4: Conv1D | input: (None, 100, 400) | output: (None, 96, 128)
max_pooling1d_1: MaxPooling1D | input: (None, 99, 128) | output: (None, 19, 128)
max_pooling1d_2: MaxPooling1D | input: (None, 98, 128) | output: (None, 19, 128)
max_pooling1d_3: MaxPooling1D | input: (None, 97, 128) | output: (None, 19, 128)
max_pooling1d_4: MaxPooling1D | input: (None, 96, 128) | output: (None, 19, 128)
concatenate_1: Concatenate | input: [(None, 19, 128), (None, 19, 128), (None, 19, 128), (None, 19, 128)] | output: (None, 76, 128)
batch_normalization_1: BatchNormalization | input: (None, 76, 128) | output: (None, 76, 128)
conv1d_5: Conv1D | input: (None, 76, 128) | output: (None, 72, 128)
max_pooling1d_5: MaxPooling1D | input: (None, 72, 128) | output: (None, 14, 128)
batch_normalization_2: BatchNormalization | input: (None, 14, 128) | output: (None, 14, 128)
conv1d_6: Conv1D | input: (None, 14, 128) | output: (None, 10, 128)
max_pooling1d_6: MaxPooling1D | input: (None, 10, 128) | output: (None, 2, 128)
batch_normalization_3: BatchNormalization | input: (None, 2, 128) | output: (None, 2, 128)
flatten_1: Flatten | input: (None, 2, 128) | output: (None, 256)
dense_1: Dense | input: (None, 256) | output: (None, 64)
batch_normalization_4: BatchNormalization | input: (None, 64) | output: (None, 64)
dense_2: Dense | input: (None, 64) | output: (None, 1)

Figure 1. The architecture of Convolutional Neural Network model for basic sentiment analysis of comments

Let $x_i \in R^k$ (where k is the dimension of the vector corresponding to the i-th word) denote a sentence with n words:

$$x_{i:n} = \; x_1 \oplus x_2 \oplus x_3 \oplus \ldots . \oplus x_n \tag{1}$$

With $\oplus$ being the convolution operator. In summary, $x_{i:i+j}$ represents the concatenation of the words $x_i, x_{i+1}, \ldots, x_{i+j}$. The convolution operator involves a filter $w \in \; R^{hk}$ which is applied to a window of h words to generate a new feature. For instance, feature c_i is created from a window of words $x_{i:i+h-1}$:

$$c_i = \; f(w.x_{i:i+h-1} + \; b) \tag{2}$$

In the equation 2, b is the bias coefficient, and f is the non-linear function. By applying a window of words to the sentence $\{x_{1:h}, \; x_{2:h+1}, \ldots, \; x_{n-h+1:n}\}$, a feature map is obtained.

$$c = [\, c_1, \; c_2, \ldots, \; c_{n-h+1}] \tag{3}$$

Given $c \in \; R^{n-h+1}$, a max-overtime pooling is performed over the feature map, taking the maximum values: $\hat{c} \in \max c$, with the idea of extracting the most important characteristics of the object. These features form the penultimate layer and are passed to the fully-connected softmax layer with the output being a probability distribution over labels. The model uses multiple filters (with different window sizes) to extract multiple features.

In order to enhance the training efficiency of this built model, additional Batch Normalization layers are utilized to normalize the features, which represent the output of each layer after activation and mitigate the non-zero-mean state. The non-zero-mean phenomenon occurs when data is not evenly distributed around the value of zero, instead having a majority of values that are either greater or less than zero. This, along with the high variance problem, leads to the data having many components that are either extremely large or extremely small. Such issues are prevalent in training neural networks with dense layers. The skewed feature distributions, characterized by outlier values that are either too large or too small, can significantly impact the network optimization process.

The utilization of Residual blocks in the proposed model has yielded promising results compared to not using them. However, as mentioned in Section 2, Convolutional Neural Network models cannot ensure the order of words in a sentence, leading to incorrect predictions. In the next section, this study aims to construct a model based on the error part, which is a Recurrent Neural Network (RNN), to further improve the accuracy of the model.

3.2 Recurrent Neural Network (RNN) model with attention mechanism for hierarchical text classification

During this phase, the aim is to construct a sentiment classification model based on variants of Recurrent Neural Networks (RNN), specifically the Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM). Moreover, while exploring the text processing pipeline, it is observed that units are processed in a specific order, starting from the smallest to the largest, namely character - word - sentence - line - paragraph. This insight has

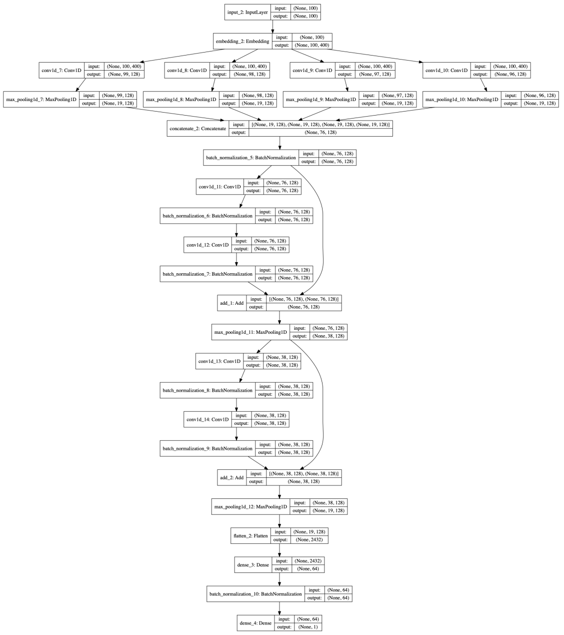

input_2: InputLayer | input: (None, 100) | output: (None, 100)
embedding_2: Embedding | input: (None, 100) | output: (None, 100, 400)
conv1d_7: Conv1D | input: (None, 100, 400) | output: (None, 99, 128)
conv1d_8: Conv1D | input: (None, 100, 400) | output: (None, 98, 128)
conv1d_9: Conv1D | input: (None, 100, 400) | output: (None, 97, 128)
conv1d_10: Conv1D | input: (None, 100, 400) | output: (None, 96, 128)
max_pooling1d_7: MaxPooling1D | input: (None, 99, 128) | output: (None, 19, 128)
max_pooling1d_8: MaxPooling1D | input: (None, 98, 128) | output: (None, 19, 128)
max_pooling1d_9: MaxPooling1D | input: (None, 97, 128) | output: (None, 19, 128)
max_pooling1d_10: MaxPooling1D | input: (None, 96, 128) | output: (None, 19, 128)
concatenate_2: Concatenate | input: [(None, 19, 128), (None, 19, 128), (None, 19, 128), (None, 19, 128)] | output: (None, 76, 128)
batch_normalization_5: BatchNormalization | input: (None, 76, 128) | output: (None, 76, 128)
conv1d_11: Conv1D | input: (None, 76, 128) | output: (None, 76, 128)
batch_normalization_6: BatchNormalization | input: (None, 76, 128) | output: (None, 76, 128)
conv1d_12: Conv1D | input: (None, 76, 128) | output: (None, 76, 128)
batch_normalization_7: BatchNormalization | input: (None, 76, 128) | output: (None, 76, 128)
add_1: Add | input: [(None, 76, 128), (None, 76, 128)] | output: (None, 76, 128)
max_pooling1d_11: MaxPooling1D | input: (None, 76, 128) | output: (None, 38, 128)
conv1d_13: Conv1D | input: (None, 38, 128) | output: (None, 38, 128)
batch_normalization_8: BatchNormalization | input: (None, 38, 128) | output: (None, 38, 128)
conv1d_14: Conv1D | input: (None, 38, 128) | output: (None, 38, 128)
batch_normalization_9: BatchNormalization | input: (None, 38, 128) | output: (None, 38, 128)
add_2: Add | input: [(None, 38, 128), (None, 38, 128)] | output: (None, 38, 128)
max_pooling1d_12: MaxPooling1D | input: (None, 38, 128) | output: (None, 19, 128)
flatten_2: Flatten | input: (None, 19, 128) | output: (None, 2432)
dense_3: Dense | input: (None, 2432) | output: (None, 64)
batch_normalization_10: BatchNormalization | input: (None, 64) | output: (None, 64)
dense_4: Dense | input: (None, 64) | output: (None, 1)

Figure 2. The architecture of Convolutional Neural Network model with built-in Residual classifier for sentiment analysis of comments

furnished the inspiration to develop a sentiment classification model that focuses on components at different levels of the text to comprehend the meaning of comments. This model is anticipated to handle lengthy, intricate comments with a high level of semantic complexity.

The aim is to use the attention mechanism applied at the word level and extends the use of context to synthesize sentence vectors. This research is conducted in stages to examine whether the attention at different text levels has different accuracy in predicting long and complex emotional texts. Notably, during the data processing process, the dots ”.” are not eliminated as usual but are used to demarcate constituent sentences in a long comment.

3.2.1 Recurrent Neural Network (RNN) model with word-level attention mechanism

Assuming that a document has L sentences s_i, where each sentence comprises T_i words. Let w_{it} represent the word at position i with $t \in [1, T]$. The proposed model processes the raw document into a vector representation, on which a classification algorithm is constructed to classify the document.

Given a sentence with the word w_{it}, $t \in [1, T]$, first, the words are embedded into a vector using an embedding matrix W_e,$x_{ij} = W_e w_{ij}$. The model employs a bidirectional Gated Recurrent Unit (GRU) to acquire word annotations by integrating information from both forward and backward directions, resulting in a more comprehensive representation of context for each word in the annotation section. Bidirectional GRU propagates to GRU $\overrightarrow{f}$ to read sentence s_i from w_{i1} to w_{iT}, and propagates backwards to $\overleftarrow{f}$ to read from w_{iT} to w_{i1}.

$$x_{it} = \ W_e w_{it.} \qquad ,t \in [1, T] \tag{4}$$

$$\overrightarrow{h_{lt}} = \overrightarrow{GRU}\,(x_{it}) \ \ , \, t \in [1, T] \tag{5}$$

$$\overleftarrow{h_{lt}} = \overleftarrow{GRU}\,(x_{it}) \ \ , \, t \in [T, 1] \tag{6}$$

To obtain an annotation for the word w_{it}, the two states $\overrightarrow{h_{lt}}$ and $\overleftarrow{h_{lt}}$ are concatenated to synthesize information surrounding the word w_{it}.

$$h_{it} = [\overrightarrow{h_{lt}}, \overleftarrow{h_{lt}}] \tag{7}$$

The attention mechanism is predicated on the notion that not all words contribute equally to convey the meaning of a sentence. Therefore, this research proposes an attention mechanism that selectively extracts words that are crucial to the meaning of the sentence, and synthesizes their representations to create a sentence vector.

$$u_{it} = \tanh\left(W_w h_{it} + \ b_w\right) \tag{8}$$

$$\alpha_{it} = \frac{\exp\left(u_{it}^T u_w\right)}{\sum_t \exp\left(u_{it}^T u_w\right)} \tag{9}$$

$$s_i \ = \sum_t \alpha_{it} h_{it} \tag{10}$$

The first step involves obtaining the annotation vector by passing it through an MLP (Multi-Layer Perceptron) layer, yielding u_{it}, the representation of h_{it}. Next, the importance of a word is measured as the similarity between u_{it} and the word-level context vector u_w, and the normalized weights α_{it} are derived via the softmax function. The sentence vector s_i is then computed as a weighted sum of the annotation vectors, with the sentence-level context vector u_w randomly initialized and jointly learned during training. In this context, document vector v serves to summarize all of the information related to the sentence within the document.

Through experiments, the proposed attention-based word-level model achieves improved accuracy as expected, compared to the baseline model. Specifically, the F1 score improved by 1.39%, and the accuracy on the test set increased by 0.8%. These results demonstrate the effectiveness of the proposed feature extraction method using the attention mechanism. The heat map, as depicted in Figure 3, provides insights into the regions of focus for the model in making its predictions.

However, in Figure 4, it can be seen that for short reviews, the model manages to capture all the key factors contributing to a high score, but for longer reviews, the model still cannot capture all the relevant factors.

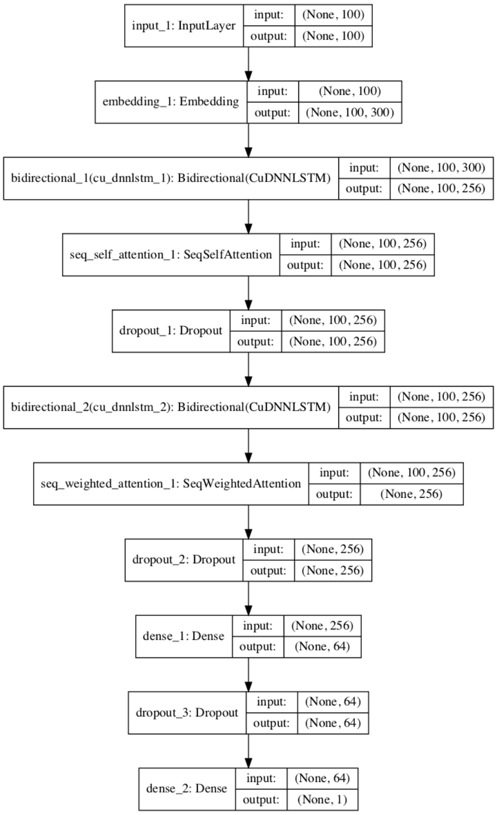
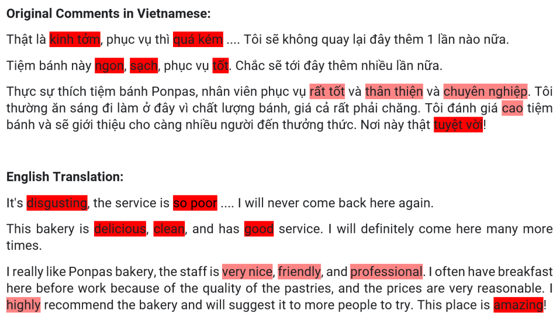


Figure 3. Architecture of the word-level attention mechanism



Original Comments in Vietnamese:

Thật là kinh tởm, phục vụ thì quá kém .... Tôi sẽ không quay lại đây thêm 1 lần nào nữa.

Tiệm bánh này ngon, sạch, phục vụ tốt. Chắc sẽ tới đây thêm nhiều lần nữa.

Thực sự thích tiệm bánh Ponpas, nhân viên phục vụ rất tốt và thân thiện và chuyên nghiệp. Tôi thường ăn sáng đi làm ở đây vì chất lượng bánh, giá cả rất phải chăng. Tôi đánh giá cao tiệm bánh và sẽ giới thiệu cho càng nhiều người đến thưởng thức. Nơi này thật tuyệt vời!

English Translation:

It's disgusting, the service is so poor .... I will never come back here again.

This bakery is delicious, clean, and has good service. I will definitely come here many more times.

I really like Ponpas bakery, the staff is very nice, friendly, and professional. I often have breakfast here before work because of the quality of the pastries, and the prices are very reasonable. I highly recommend the bakery and will suggest it to more people to try. This place is amazing!

Figure 4. Heat map of the contribution level of words in sentence's emotional rating

For instance, in the third review sentence, it seems that the model allocates much attention to the word ”tuyet voi” (amazing) and pays less attention to other important words such as ”rat tot” (very nice) and ”than thien” (friendly). In the following section, the attention mechanism of the model will be extended to address this limitation. The objective is to accurately detect sentiments in lengthy and intricate review sentences.

3.2.2 Recurrent Neural Network (RNN) model with word-and-sentence-level attention mechanism

This proposed model is based on the idea that a document is composed of multiple sentences, and that directing attention to each sentence can capture the meaning of the document as a whole. Consequently, through the input of sentence vectors s_i, a document vector can be derived in that way. A bidirectional LSTM network is utilized to encode the sentences.

$$\overrightarrow{h_l} = \overrightarrow{LSTM}(s_i), i \in [1, L] \tag{11}$$

$$\overleftarrow{h_l} = \overleftarrow{LSTM}(s_i), i \in [L, 1] \tag{12}$$

The sentence's annotation vector is obtained through the concatenation of $\overrightarrow{h_l}$ and $\overleftarrow{h_l}$.

$$h_i = [\overrightarrow{h_l}, \overleftarrow{h_l}] \tag{13}$$

The vector h_i synthesizes neighboring sentences while still focusing on sentence i. The purpose of word-and-sentence-level attention is to pinpoint essential sentences for the purpose of document classification, thereby providing the dominant sentiment of the entire text. In this study, a sentence-level context vector u_s is introduced to gauge the significance of each sentence in the document.

$$u_i = \tanh\left(W_s h_i + b_s\right) \tag{14}$$

$$\alpha_i = \frac{\exp\left(u_i^T u_s\right)}{\sum_t \exp\left(u_i^T u_s\right)} \tag{15}$$

$$v = \sum_t \alpha_i h_i \tag{16}$$

The document vector v summarizes all of the information contained in the document's sentences. Likewise, sentence-level context vectors u_s may be randomly initialized and jointly learned during the training process, as indicated in Figure 5.

4 Experiments and Results

4.1 Overall proposed process for sentiment classification of Vietnamese comments

The overall process of this experiment for sentiment classification of Vietnamese comments has been depicted in Figure 6.

The first phase involved data collection from different sources, followed by a comprehensive preprocessing step that involved the rectification of misspelled words, acronyms, icons, and standardization of punctuations, as well as the segmentation of words. In the subsequent stage, the documents were vectorized utilizing pretraining and labeling data. Finally, the dataset was segregated into training, validating, and testing sets; and the proposed model was run to generate the evaluation results.

Table 1. Descriptive statistics of the training dataset

Source	Manually collected	VLSP 2016	Total
Number of Comments	10084	6003	16087
Percentage	62.7%	37.3%	100.00%
Positive: negative ratio	55:45	63:37	57:43

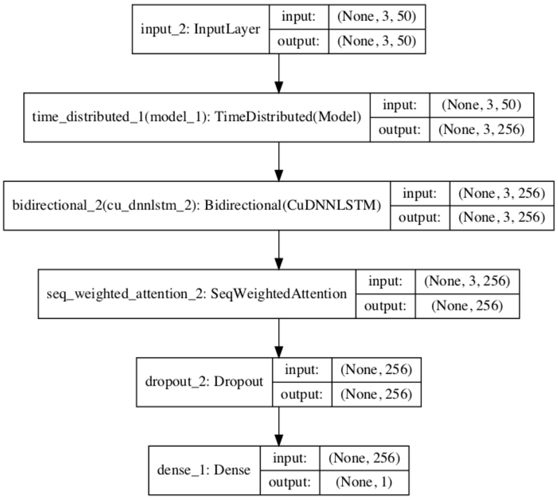


Figure 5. Architecture of the word-and-sentence-level attention mechanism

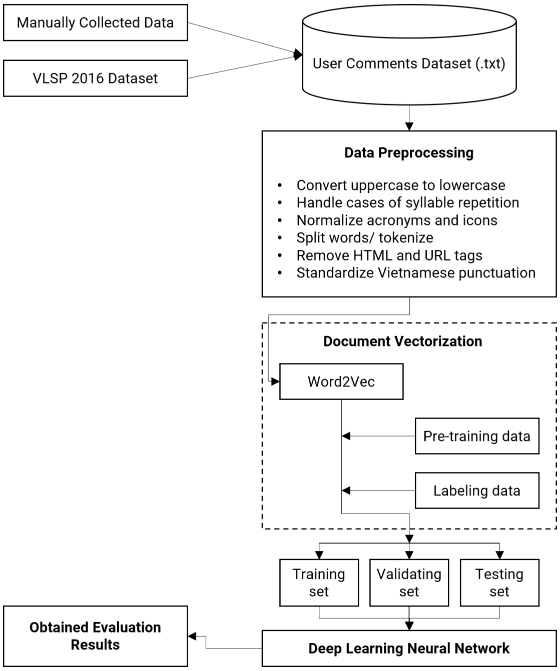

Manually Collected Data
VLSP 2016 Dataset
User Comments Dataset (.txt)
Data Preprocessing
• Convert uppercase to lowercase
• Handle cases of syllable repetition
• Normalize acronyms and icons
• Split words/ tokenize
• Remove HTML and URL tags
• Standardize Vietnamese punctuation
Document Vectorization
Word2Vec
Pre-training data
Labeling data
Training set
Validating set
Testing set
Obtained Evaluation Results
Deep Learning Neural Network

Figure 6. Overall proposed process for sentiment classification of Vietnamese comments

4.1.1 Data Collection

The dataset was first collected from the Foody (`https://www.foody.vn/`) food ordering application, which is a search and review application for restaurants in Vietnam. Manual labeling was performed by the research team to classify the data as positive or negative sentiment. Furthermore, the dataset was supplemented with data extracted from the pre-labeled VLSP 2016 dataset [Nguyen et al., 2018]. Details of the generated dataset are provided in Table 1. Upon analyzing the dataset, it was observed that:

- For the manually collected dataset, the comments were relatively short, with a significant amount of icons and abbreviations being used.
- For the VLSP 2016 dataset, the comments were comparatively longer, with well-formatted and accurately spelled data. A majority of the comments were related to technology products such as computers and phones.

As can be observed from Table 1, approximately two-thirds of the final dataset were manually collected, with the remaining one-third sourced from the VLSP 2016 dataset. Moreover, the merged dataset has a ratio of positive and negative sentiment comments of 57:43.

4.1.2 Data Preprocessing

For this sentiment analysis task, either positive or negative comments were used, while neutral comments were discarded as they did not contribute to sentiment classification. Based on the dataset analysis, the following data preprocessing steps were applied:

- Convert uppercase letters to lowercase: ”Ngon” becomes ”ngon” (delicious);
- Handle cases of syllable repetition: ”ngooon quaaa điiiii” becomes ”ngon qua đi” (so delicious);
- Normalize abbreviations and icons: ”k”, ”ko”, ”k0” become ”khong” (not), ”bt” becomes ”bình thuong” (normal);
- Split the words/ tokenize: using the spaCy tokenization library [xx];
- Remove HTML and URLs tags;
- Standardize Vietnamese punctuation.

4.1.3 Document Vectorization

Typically, computers cannot understand the meanings of words. Therefore, to process natural language, a method of representing text in a form that computers can understand is needed. The standard method of representing text is through vector representations. This study uses the pre-trained word2vecVN model [Vu et al., 2018] [Le et al., 2021] trained using the skip-gram method [Mikolov et al., 2013] on a dataset of 7.1GB of Vietnamese text, 1,675,819 unique vocabularies

Table 2. Comparison of the accuracy and F1 scores of all the models on the training and testing datasets

Model	Accuracy	F1 score on the training set	F1 score on the testing set
CNN	91.89%	86.26%	91.85%
CNN with Residual Blocks	92.63%	88.35%	92.56%
RNN with word-level attention mechanism	93.41%	89.74%	93.37%
RNN with word-and-sentence-level attention mechanism	92.74%	89.45%	92.57%

extracted from 974,393,244 raw vocabulary and 97,440 text files.

The result obtained after vectorization with word2vec is a dictionary in which each vocabulary is represented as a 400-dimensional vector, as presented in Figure 7. Each word in the dataset's sentences is then replaced with the corresponding vector in the dictionary. The dataset is divided into three parts: 60% of the total comments are used for training, 20% for validation, and the remaining 20% for testing.

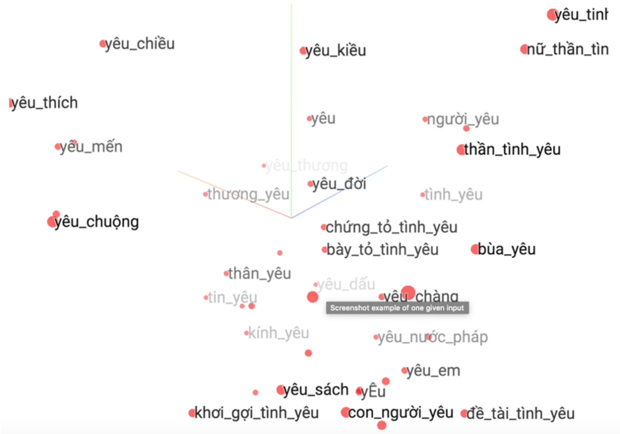


Figure 7. Representing vocabulary on the space using word2vecVN model

4.2 Results and Discussion

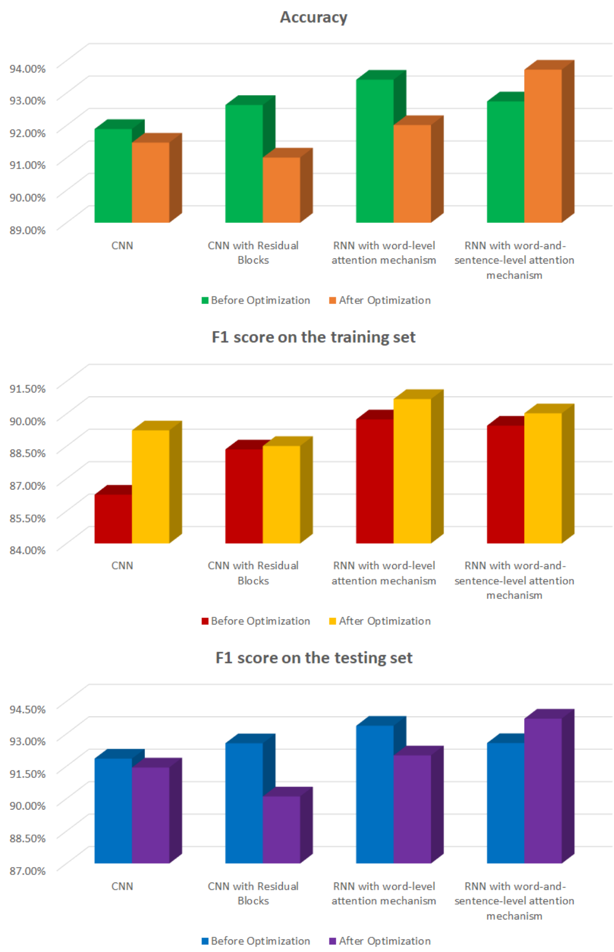
In the task of imbalanced classification, where the dataset of classes varies significantly, the presented models were experimented with and evaluated using the F1 score metric [Goutte et al., 2005]. Higher values of these metrics indicate more effective classification. Subsequently, the accuracy index is employed to evaluate the model on both the training and testing sets.

Table 2 clearly demonstrates a notable improvement in accuracy and F1 score on both the training and testing datasets of the word-level and sentence-and-word-level attention model, compared to the CNN and the CNN with Residual Blocks model. The RNN model with word-level attention achieved the highest F1 score of 93.37% on the testing set, while the RNN model with word-and-sentence-level attention reached a high F1 score on the testing set of 92.57%.

This result also shows that the accuracy of the word-level attention model is higher than the word-and-sentence-level attention model. This can be explained by the fact that most of the comments in the manually labeled dataset are two to three sentences long with complex structures. Hence, the word-and-sentence-level attention model presently faces a challenge in detecting lengthy sentences, although it maintains stability and has a high rate of accurate prediction. Therefore, in the subsequent section, this study puts forward an optimization strategy that involves substituting words with comparable meanings to overcome this constraint.

4.3 Word-and-sentence-level Attention Model Optimization: Replacing Words with Similar Meanings

In order to enhance the accuracy of the word-and-sentence-level attention model, the task of replacing words with similar meanings was carried out. The semantic similarity was calculated by measuring the similarity of two vectors in space. Then, the values of similarity greater than 0.5 (threshold ≥ 0.5) were taken, and the words with similar meanings were replaced in the original dataset. This updated dataset was then added to the original dataset, resulting in a 1.5-fold increase in the size of the training dataset. Figure 8 demonstrates the effectiveness of this method of replacing synonymous words, which significantly improves the accuracy and F1 score of the word-and-sentence-level attention model. Table 3 and Figure 8 compare the accuracy and F1 scores of all the models after replacing words with similar meanings. They demonstrate the effectiveness of the method of replacing words in the same context to enrich the dataset, resulting in improved accuracy of the word-and-sentence-level attention model, with the highest F1 score of 93.7% in the testing set. However, it is also discovered that the replacing words with similar meanings technique is not suitable for other models (CNN, CNN with Residual Blocks, RNN with word-level attention mechanism) as these models evaluate based on the meaning of all sentiment words, not the overall tone of each sentence in a document. Therefore, this optimization technique should only be applied

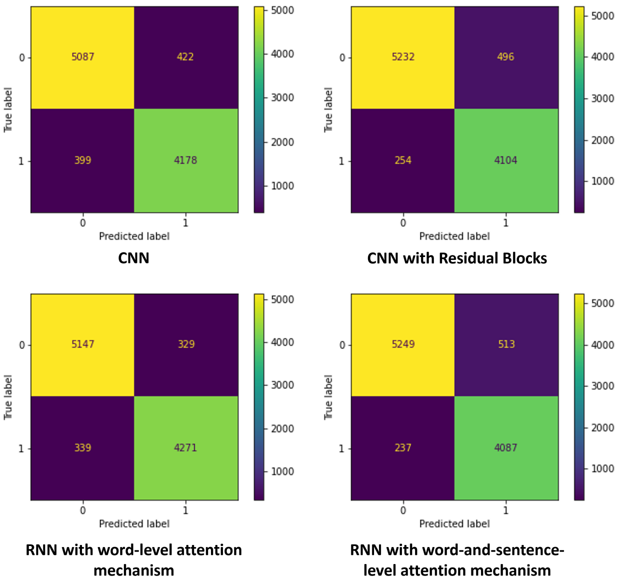


Accuracy

F1 score on the training set

F1 score on the testing set

Figure 8. Comparison of F1 score and accuracy of all the models before and after replacing words with similar meanings



CNN

CNN with Residual Blocks

RNN with word-level attention mechanism

RNN with word-and-sentence-level attention mechanism

Figure 9. Confusion Matrix of all models

Table 3. Comparison of the accuracy and F1 scores of all the models after replacing words with similar meanings

Model	CNN	CNN with Residual Blocks	RNN with word-level attention mechanism	RNN with word-and-sentence-level attention mechanism
Accuracy	91.47%	91.01%	92.01%	93.72%
F1 score on the training set	89.23%	88.51%	90.68%	90.02%
F1 score on the testing set	91.44%	90.1%	92%	93.7%
Precision	91.77%	92.29%	93.33%	92.26%
Recall	91.81%	92.76%	93.31%	92.80%

to the RNN with the word-and-sentence-level attention mechanism model. Moreover, the precision and recall values have yielded commendable results, while Figure 9 illustrates the corresponding confusion matrices.

5 Conclusions and Future Work

This study conducts a series of experiments for sentiment classification of Vietnamese comments. The objective of the research is to assess the emotions conveyed in comments on e-commerce websites and classify them as either negative (N) or positive (P). Unlike the majority of existing studies that have focused on the English language, this research aims to process Vietnamese comments, on a dataset manually collected from the Foody app and the VLSP 2016 dataset. Despite substantial efforts in the past [Liu, 2020],[Hu et al., 2020],[He et al., 2016], the study proposed Recurrent Neural Network (RNN) models with word-and-sentence-level attention mechanisms and enhanced the accuracy by replacing words with similar meanings. With impressive results for the accuracy of 93.72% and F1 score of 93.7%, the proposed model significantly outperformed the traditional CNN model and CNN with the Residual Blocks model. These results are promising and could contribute to the field of text classification by gaining a deeper understanding of the hidden meanings in complex text.

Future work aims to improve the accuracy of sentiment analysis by enriching the dataset, using more powerful word segmentation methods, and embedding words with more contexts. Moreover, instead of only classifying sentiments into two labels, ”positive” and ”negative”, future work aims to classify them into six labels: happy, sad, angry, surprised, disgusted, fearful; or more. Additionally, the model needs to be continuously updated as common abbreviations increasingly appear.

Another direction for research is to expand the comment analysis model to many different languages. With a good quality foreign language dataset, effective word segmentation and word embedding methods combined with the models above can yield results similar to the Vietnamese comment classification model that this study has developed. Specifically in this project, the authors have also extended the research scope to analyze Japanese comments. With the word segmentation method using the Janome library, the word2vec dataset is pre-trained with each word's dimension being 300, the proposed word-and-sentence-level attention model above yields very promising results with an F1 score of 91.76%.

In summary, this research contributes to the field of sentiment classification in texts written in complex languages such as Vietnamese and Japanese. The model could be applied in various fields such as opinion mining, customer feedback analysis, and natural language processing. Moreover, high-accuracy sentiment classification models can be applied in practical scenarios in e-commerce businesses, where shop owners can observe the customers' response attitudes and make appropriate adjustments.

References

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*

Graves, A., Mohamed, A-R., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *IEEE international conference on acoustics, speech and signal processing*, pp. 6645-6649.

Bengio, Y., and Senecal, J-S. (2003). Adaptive importance sampling to accelerate training of a neural probabilistic language model. *IDIAP, Tech. Rep., speech and signal processing*

Yih, W-T., Toutanova, K., Platt, J.C., and Meek, C.(2011). Learning discriminative projections for text similarity measures. In *Proceedings of the fifteenth conference on computational natural language learning*, pp. 247-256.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, vol. **26**.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, pp. 2493–2537.

Zhang, C., Li, Q., and Song, D. (2019). Aspect-based sentiment classification with aspect-specific graph convolutional networks. *arXiv preprint arXiv:1909.03477*

Liu, D. (2020). Low-Latency End-to-End Speech Recognition with Enhanced Readability. In *PhD Thesis, Maastricht University*.

Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. (2014). Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd international conference on world wide web*, pp. 373–374.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modeling sentence. *arXiv preprint arXiv:1404.2188*

Hu, H., Liao, M., Zhang, C., and Jing, Y. (2020). Text classification based recurrent neural network. *IEEE 5th Information Technology and Mechatronics Engineering Conference*, pp. 652–655.

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. In *Neural computation*, vol. **9**(8), pp. 1735–1780.

Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*

Hien, N., L., H., Hoang, H., M., Tien, N. V., and Hieu, N., V. (2021). Keyphrase extraction model: a new design and application on tourism information. *Informatica*, vol. **45**(4).

Im Tan, L., San Phang, W., Chin, K. O., and Patricia, A. (2015). Rule-based sentiment analysis for financial news. *IEEE 2015 International Conference on Systems*, pp. 1601-1606.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Rule-based sentiment analysis for financial news. In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631-1642.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778.

Hien, N., L., H.(2022). Deep Learning Analysis and Optimization for Different Data Sources in Autonomous Vehicles.

Mnih, V., Heess, N., Graves, A., et al. (2014). Recurrent models of visual attention. Advances in neural information processing systems. *Advances in neural information processing systems*, vol. **27**.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*

Goutte, C., and Gaussier, E. (2005). A probabilistic interpretation of precision, recall and F-score, with implications for evaluation. In: *ECIR Advances in In-*

formation Retrieval: 27th European Conference on IR Research, pp. 345–359

Nguyen, H. T. M., Nguyen, H. V., Ngo, Q. T., Vu, L. X., Tran, V. M., Ngo, B. X., and Le, C. A. (2018). VLSP shared task: sentiment analysis. *Journal of Computer Science and Cybernetics*, vol. **34**(4), pp. 295–310.

Vu, T., Nguyen, D. Q., Nguyen, D. Q., Dras, M., and Johnson, M. (2018). VnCoreNLP: A Vietnamese natural language processing toolkit. *arXiv preprint arXiv:1801.01331*

Le, A. P., Pham, T. V., Le, T. V., and Huynh, D. V. (2021). Neural Transfer Learning For Vietnamese Sentiment Analysis Using Pre-trained Contextual Language Models. In: *ICMLANT 2021 IEEE International Conference on Machine Learning and Applied Network Technologies*, pp. 1–5.

Hien, N. L. H., Tien, T. Q., and Hieu, N. V. (2020). Web crawler: Design and implementation for extracting article-like contents. *Cybernetics and Physics*, vol. **9**(3), pp. 144–151.

Jain, V. K., and Kumar, S. (2017). Improving customer experience using sentiment analysis in e-commerce. In: *Handbook of Research on Intelligent Techniques and Modeling Applications in Marketing Analytics*, vol. **9**(3), pp. 216–224.

Ameur, H., and Jamoussi, S. (2013). Dynamic construction of dictionaries for sentiment classification. In: *IEEE 13th International Conference on Data Mining Workshops*, pp. 3896–3903

Bazhanov, P., Kotina, E., Ovsyannikov, D., and Ploskikh, V. (2018). Optimization algorithm of the velocity field determining in image processing. *Cybernetics And Physics*. **7**, pp. 174–181.

Shats, V. (2022). Properties of the Ordered Feature Values as a Classifier Basis. *Cybernetics And Physics*. **11**, pp. 25–29.

Habib, R., Hamed, B., Siavash, S., Ahmad, K., Yaser, K., and Yahya, T. (2022). Incremental deep learning training approach for lesion detection and classification in mammograms. *Cybernetics And Physics*. **11**, pp. 234–245.