Vietnamese sentiment analysis

December 2023, HUST
Deep Learning

Pham Duc Thanh
20210795
Vu Nhat Minh
20214919
Bui Anh Nhat
20210657
Tran Van Toan
20214932
Dao Ha Xuan Mai
20210562
Ta Quang Duy
20214884

Table of Contents

Abstract

Sentiment analysis serves as a potent approach for extracting valuable insights from reviews, categorizing them based on positive or negative sentiments. In the context of the UIT-VSMEC (Vietnamese Sentiment Emotion) Project, our goal is to employ Machine Learning and Deep Learning methodologies to assess the model's accuracy and determine the optimal algorithm for sentiment analysis.

1 Introduction

In the era of the internet and rapid technological advancement, the development of intelligent systems has gained substantial momentum, aiming to automate tasks traditionally performed by humans. Within the domain of Natural Language Processing (NLP), a particularly intriguing challenge has emerged: categorizing text-based comments into distinct emotional categories. Unlike traditional sentiment analysis, which classifies text as positive, negative, or neutral, this project delves into the more nuanced realm of emotion recognition. Emotion analysis plays a pivotal role in understanding the sentiment and underlying emotions expressed in text, offering valuable insights across various applications, from assessing user reactions to content to understanding customer feedback in depth. To tackle the task of categorizing comments into emotions, we employ a diverse array of machine learning and deep learning models. These models are designed to harness the power of multi-layered neural networks, allowing them to learn intricate feature representations from the training data and develop predictive models for emotional categorization. However, our primary focus lies on the latest deep learning techniques, which have demonstrated remarkable capabilities in capturing the nuanced nuances of human emotions expressed in text. Central to our project is the UIT-VSMEC (University of Information Technology - Vietnamese Sentiment Emotion Corpus) dataset, meticulously designed for the Vietnamese language. This dataset serves as the cornerstone for training and evaluating our machine learning and deep learning models, providing them with a rich source of labeled data to learn from. Our endeavor represents a pioneering effort in the realm of emotion analysis for Vietnamese text, aimed at not only understanding the sentiment but also decoding the intricate tapestry of emotions expressed by users. Through the fusion of cutting-edge technology and language-specific dataset, we embark on a journey to advance emotion recognition and bring deeper insights to the world of text-based communication.

2 Dataset

2.1 Data set

Vietnamese Social Media Emotion Corpus (UIT-VSMEC) has about 6,927 human-annotated sentences with six emotion labels (**sadness, enjoyment, anger, disgust, fear and surprise**), contributing to emotion recognition research in Vietnamese which is a low-resource language in Natural Language Processing (NLP).

Based on Ekman's instruction in basic human emotions, they classify for Vietnamese text with seven emotion labels described as follows:

- Enjoyment: For comments with the states that are triggered by feeling connection or

sensory pleasure. It contains both peace and ecstasy. The intensity of these states varies from the enjoyment of helping others, a warm uplifting feeling that people experience when they see kindness and compassion, an experience of ease and contentment or even the enjoyment of the misfortunes of another person to the joyful pride in the accomplishments or the experience of something that is very beautiful and amazing. For example, the emotion of the sentence "Nháy mt thôi cũng đáng yêu, kkk" (English translation: "Just the act of winking is so lovely!") is Enjoyment.

- Sadness: For comments that contain both disappointment and despair. The intensity of its states varies from discouragement, distraughtness, helplessness, hopelessness to strong suffering, a feeling of distress and sadness often caused by a loss or sorrow and anguish. The Vietnamese sentence "Lúc đy kh lm... k nim :(" (English translation: "It was hard that time..memory :(") has an emotion of Sadness, for instance.

- Fear: For comments that show anxiety and terror. The intensity of these states varies from trepidation - anticipation of the possibility of danger, nervousness, dread to desperation, a response to the inability to reduce danger, panic and horror - a mixture of fear, disgust and shock. A given sentence "Chuyn này làm tao ni ht da gà" (English translation: "This story causes me goosebumps") is a Fear labeled-sentence

- Anger: For comments with states that are triggered by a feeling of being blocked in our progress. It contains both annoyance and fury and varies from frustration which is a response to repeated failures to overcome an obstacle, exasperation - anger caused by strong nuisance, argumentativeness to bitterness - anger after unfair treatment and vengefulness. For example, "Bin m mày đi!" (English translation: "You fucking get lost!") is labeled with Angry.

- Disgust: For comments which show both dislike and loathing. Their intensity varies from an impulse to avoid something disgusting or aversion, the reaction to a bad taste, smell, thing or idea, repugnance to revulsion which is a mixture of disgust and loathing or abhorrence - a mixture of intense disgust and hatred. As "Làm bn vi my th loi này nhc c ngi" (English translation: "Making friends with such types humiliates you") has an emotion of Disgust.

- Surprise: For comments that express the feeling caused by unexpected events, something hard to believe and may shock you. This is the shortest emotion of all emotions, only takes a few seconds. And it passes when we understand what is happening, and it may become fear, anger, relief or nothing ... depends on the event that makes us surprise. "Trên đi còn tn ti th này sao??" (English translation: "How the hell in this world this still exists??") is annotated with Surprise

- Other: For comments that show none of those emotions above or comments that do not contain any emotions. For instance, "Mình đã xem rt nhiu video nh này ri nên thy cũng bình thng" (English translation: "I have seen a lot of videos like this so it's kinda normal") is neutral, so its label is Other.

2.2 Data preproccessing

As with any Machine Learning or Deep Learning approach, a processing procedure is required to prepare raw data for use in model training and testing processes.

Our process includes:

- Lowercase and handle punctuations: Lowercase is a common technique when preprocessing text data. It helps the model to understand correctly about the words. For the punctuations handling, we have run many tests on which punctuations to keep and decided to keep the

comma, dots, question mark, exclamation point and remove others. We believe these punctuations can provides valuable sentiment information for our models (eg: exclamation point can express surprise)

- Remove duplicate characters in words, and words in sentences such as đepppppppp quáaa.

- replace special words in Vietnamese to their original meaning: 'ô kêi': ' ok ', 'o kê': ' ok ', 'kh ':' không ', 'kô ':' không ', 'hok ':' không ',...

- Handle English words and too long words: In Vietnamese, the longest word is "nghiêng" (inclined), which contains 7 characters, so any words that is more than 7 characters will indicate that it is misspelled. But in our dataset, there are also many English words, which can be more than 7 characters. To solve this problem, we use an English dictionary to filter out the English words and remove all other words with more than 7 characters

- Handle misspelled words: In Vietnam, different regions will have different ways to pronunciate. Because of that, in some regions, people often have same mistakes in pronunciating and writing Vietnamese. For example: Many people might write "cm n" or "giá r" as "cãm n" or "giá r". To overcome this problem, we first analyse and choose the top words that are misspelled and manually created a small dictionary to correct the spelling mistakes.

- Handle emoticons and emojis: Since body language and verbal tone do not translate in our text messages or e-mails, people have developed alternate ways to convey nuanced meaning. The most prominent change to our online style has been emoticons and emoji. Emoticons are punctuation marks, letters, and numbers used to create pictorial icons that generally display an emotion or sentiment. In other side, emoji are pictographs of faces, objects, and symbols. In our datasets, there are many emoji and emoticons, and how to handle them properly is crucial, as it may contain sentiment information we need. We have tested two cases: remove all emoji/emoticons and replace emoji/ emoticons by words and obtain that by replacing them by word using Emoji library, we can have better results.

2.3 TD-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is a numerical statistic that reflects the importance of a word in a document relative to a collection of documents, often used in natural language processing and information retrieval. It is a combination of two components: Term Frequency (TF) and Inverse Document Frequency (IDF).

$$TF(t,d) = \frac{\text{Number of times term t appears in document d}}{\text{Total number of terms in document d}}$$

TF measures the frequency of a term within a document. It is calculated by dividing the number of times a term appears in a document by the total number of terms in that document. The idea is to emphasize words that occur frequently within a specific document, indicating their relevance to the document's content.

On the other hand, IDF measures the uniqueness of a term across a collection of documents. It is calculated by dividing the total number of documents by the number of documents containing the term, followed by taking the logarithm of the result. The purpose is to highlight terms that are rare and distinctive, as they may carry more significance.

$$IDF(t,D) = \log(\frac{\text{Total number of documents in the collection D}}{\text{Number of documents containing term t}})$$

The TF-IDF score for a term in a specific document is obtained by multiplying its TF and IDF values:

$$TF - IDF(t, d, D) = TF(t, d) * IDF(t, D)$$

In summary, TF-IDF is a powerful tool for evaluating the importance of terms in a document within the context of a larger collection. It allows for the identification of key terms that are both frequent within a document and rare across the entire document set, thus aiding in tasks such as text mining, information retrieval, and document categorization.

2.4 Word Embedding (word2vec)

Word embedding is a term used in natural language processing (NLP) to describe a technique where words or phrases from a vocabulary are mapped to vectors of real numbers. This mapping is done in a way that words with similar meanings are close to each other in the vector space, which helps with text analysis and understanding the semantic relationships between words.

2.4.1 One-Hot Vectors

One-Hot Vectors is easy to understand and construct. Imagine we have a dictionary with N unique words, each assigned a unique integer index from 0 to N-1. To create a one-hot vector for a word with index i, we construct a vector of length N, filled with zeros, and set the i-th element to 1. Consequently, each word is represented as an N-length vector, which can be directly utilized by neural networks. However, because One-Hot Vectors cannot accurately express the similarity between different words, such as the cosine similarity that we often use, it's not usually a good choice.

2.4.2 The Skip-Gram Model

The Skip-Gram model is a straightforward neural network with a single hidden layer, designed to estimate the probability of a particular word appearing in the context of a given input word. This model works in reverse to the Continuous Bag of Words (CBOW) model. It takes the current word as input and aims to predict the words that come before and after it. In essence, it learns and predicts the context words around the input word. Performance evaluations of this model have indicated that prediction quality enhances with a larger set of word vectors, although this also escalates computational complexity.

The Skip-Gram objective function aggregates the log probabilities of the surrounding n words to the left and right of the target word wt to formulate the objective. This process can be visualized as follows:

$$J_\theta = \frac{1}{T} \sum_{t=-n \leq j \leq n, \neq 0}^{T} \log p(w_(t + j)|w_t)$$

- θ: all variables.
- n: size of training.
- T: number of words.

2.4.3 The Continuous Bag of Words Model

The Continuous Bag of Words (CBOW) model is similar to the skip-gram model, but instead of predicting the surrounding words given a center word, it predicts the center word given its surrounding context words. This means that the CBOW model takes a sequence of words as input, and then tries to predict the most likely word to appear in the next position.

To do this, the CBOW model uses a neural network that takes the vectors of the surrounding words as input, and then outputs a vector that represents the predicted center word. The neural network is trained by minimizing the distance between the predicted vector and the actual vector of the center word.

The objective function for CBOW:

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} \log p(w_t|(w_t - n, ..., w_t - 1, w_t + 1, ..., w_t + n))$$

- θ: all variables.
- T: number of words.

The CBOW model has a number of advantages over the skip-gram model. First, it is able to learn more complex relationships between words, as it is able to take into account the context of the center word. Second, it is more efficient to train, as it only needs to calculate the gradient for the center word, rather than for all of the surrounding words.

2.5 Exploratory Data Analysis

We concluded that the comments got from social network are uneven in number among different labels in which the enjoyment label reaches the highest number of 1,965 sentences (28.36%) while the surprise label arrives at the lowest number of 309 sentences (4.46%).

Emotion	**Sentences**	**Percentage** (%)
Enjoyment	1,965	28.36
Disgust	1,338	19.31
Sadness	1,149	16.59
Anger	480	6.92
Fear	395	5.70
Surprise	309	4.46
Other	1,291	18.66
Total	6,927	100

Fig. 1: Statistics of emotion labels of the UIT-VSMEC corpus

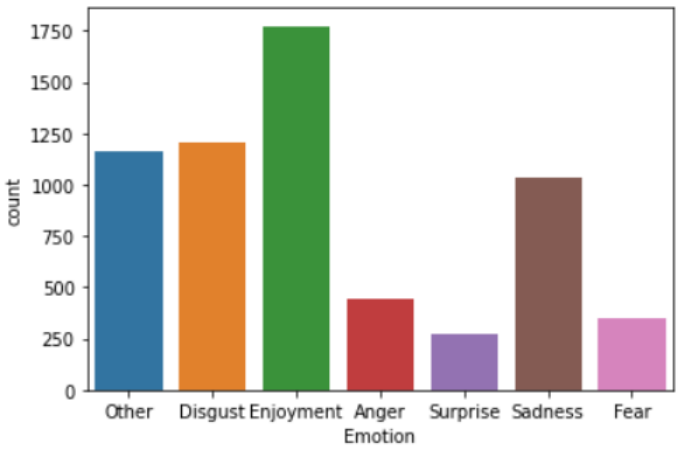

Fig. 2: Distribution of data by sentiments

Length	Enjoyment	Disgust	Sadness	Anger	Fear	Surprise	Other	Overall
1-5	5.16	2.84	1.70	0.69	0.94	0.85	1.87	**14.05**
6-10	8.98	4.22	4.98	1.41	1.42	2.25	7.20	**30.38**
10-15	5.87	3.99	4.11	1.40	1.27	0.94	5.00	**22.58**
16-20	4.17	3.05	2.51	1.14	0.85	0.24	2.79	**14.75**
21-25	1.96	1.93	1.50	0.66	0.40	0.15	1.11	7.71
26-30	1.08	1.31	0.95	0.45	0.27	0.01	0.53	4.6
>30	1.23	1.97	0.84	1.17	0.55	0.02	0.15	5.93
Total	28.36	19.31	16.59	6.92	5.70	4.46	18.66	100

Fig. 3: Distribution of emotion-annotated sentences according to the length of the sentence

3 Methodology

3.1 Machine learning methods

3.1.1 XGBoost

XGBoost, or eXtreme Gradient Boosting, is an advanced machine learning algorithm that has gained prominence for its exceptional performance in supervised learning tasks. It operates within the realm of ensemble learning, a methodology that combines the predictive strength of multiple models to enhance overall accuracy and generalization. What sets XGBoost apart is its sequential construction of decision trees, where each subsequent tree focuses on correcting the errors of the preceding ones. This process, known as boosting, allows XGBoost to incrementally refine its predictions, producing a robust and accurate model.

The algorithm's optimization hinges on a carefully crafted objective function that comprises two crucial components: a loss function and a regularization term. The loss function quantifies the disparity between predicted and actual values, while the regularization term helps prevent the model from becoming overly complex and overfitting the training data. Through an iterative process, XGBoost minimizes this combined objective function, striking a balance between precision and model simplicity. One key feature that contributes to XGBoost's popularity is its interpretability. The algorithm provides valuable insights into the importance of each feature in the dataset, offering a clear understanding of how these features influence the model's

predictions. This interpretability is especially advantageous in scenarios where understanding the underlying decision-making process is as crucial as predictive accuracy.

XGBoost's versatility is evident in its applicability to a wide range of machine learning tasks. Whether tackling classification problems, regression analyses, or ranking challenges, XGBoost has proven itself as a reliable and efficient solution. Its adaptability to diverse datasets and ability to handle large-scale, high-dimensional data make it a favored choice among data scientists and machine learning practitioners seeking powerful models for real-world applications. In essence, XGBoost stands as a sophisticated and versatile tool, making it a cornerstone in the toolkit of machine learning professionals.

3.1.2 SVM

Support Vector Machines (SVM) are a versatile class of supervised learning algorithms with a robust mathematical foundation. At their core, SVMs are designed to find the hyperplane that best separates classes in the feature space. The algorithm achieves this by identifying support vectors—data points crucial for defining the decision boundary. The concept of a margin, representing the distance between the hyperplane and the nearest data point of any class, is central to SVM. SVMs excel in scenarios where the goal is to maximize this margin, as it leads to a more resilient and generalizable model.

One notable feature of SVM is its ability to handle non-linear decision boundaries. This is accomplished through the use of kernel functions, such as radial basis function (RBF) or polynomial kernels, which implicitly map input features into a higher-dimensional space where a linear separation is feasible. Its formula:

$$K(x, z) = e^{\frac{||x-z||^2}{2\sigma}}, \text{ where } \sigma > 0$$

The regularization parameter (C) in SVM controls the trade-off between achieving a smooth decision boundary and correctly classifying training data. A smaller C value encourages a broader margin, potentially allowing for some misclassifications, while a larger C emphasizes accurate classification, potentially leading to a narrower margin.

SVMs find application in diverse domains, including image recognition, text categorization, and bioinformatics. Their efficacy in high-dimensional spaces, even when the number of features exceeds the number of samples, makes SVMs particularly suitable for tasks with complex data structures. While SVMs are powerful, their performance can be influenced by the choice of kernel and tuning parameters, requiring careful consideration in practice. Nevertheless, SVMs remain a foundational tool in machine learning, celebrated for their ability to handle various types of data and produce robust decision boundaries.

3.2 Deep learning methods

3.2.1 Convolutional neural network

This method for text classification utilizes Convolutional Neural Networks (CNNs) (Yoon Kim 2014) to extract hierarchical features from input data. In the context of Vietnamese text classification, this approach employs a pre-trained embedding model called PhoWord2Vec, specifically trained for the Vietnamese language. This embedding captures semantic relationships between words, enhancing the model's understanding of the nuances in Vietnamese text.

As part of the preprocessing pipeline, VNCoreNLP is utilized for word segmentation, ensuring accurate delineation of individual words in the text. This segmentation aids in the model's ability to comprehend the structure and context of the input, crucial for effective classification.

Instead of image pixels, the input to most NLP tasks are sentences or documents represented as a matrix. Each row of the matrix corresponds to one token, that is, each row is vector that represents a token.

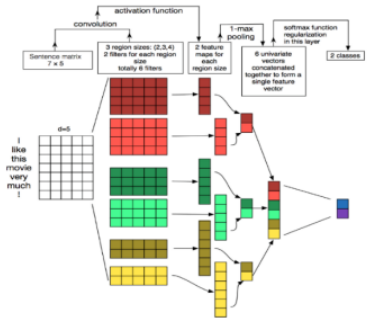

Fig. 4: Example of a CNN model for NLP

3.2.2 RNN

Leveraging Recurrent Neural Networks (RNNs) for multiclass Vietnamese text classification, this method integrates PhoWord2Vec (dim = 300) embeddings and VNCoreNLP tools for word segmentation during preprocessing. RNNs excel in capturing sequential dependencies, while PhoWord2Vec enhances semantic understanding. The use of VNCoreNLP ensures accurate word segmentation, collectively enabling the model to discern nuanced patterns in Vietnamese text for classification.

3.2.3 LSTM

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is capable of learning long-term dependencies in sequence data. This is particularly useful for tasks involving sequential inputs like natural language processing, speech recognition, and time series prediction. LSTM models overcome the limitations of traditional RNNs, such as vanishing gradient problem, by using a unique gating mechanism that controls the flow of information between cells in the network.

The LSTM layers take into account not only the word order but also their contextual significance within the sentence. Through this process, the model discerns key patterns residing within the sequences, identifying their correlation with specific emotions. The ultimate layer of the model typically consists of a softmax layer, producing a probability distribution of potential emotions. With the highest-probability emotion selected as its prediction, this model possesses a key advantage in its adaptability to varying sentence lengths due to the inherent properties of LSTM networks. As such, it boasts immense versatility and efficacy in accurately classifying the conveyed emotion.

3.2.4 GRU

Introduced in 2014 by Kyunghyun Cho et al., GRU has established itself as a well-known type of recurrent neural network. It was designed as a simpler alternative to Long Short-Term Memory (LSTM) networks, with fewer parameters, making it more computationally efficient.

GRU has gained popularity in the field of deep learning, particularly in tasks involving sequential data like natural language processing, speech recognition, and time-series prediction. Its simpler architecture compared to LSTM makes it faster to train, which can be advantageous in projects where computational resources or time are limiting factors.

However, it's important to note that while GRU has been widely adopted, LSTM is still more prevalent in certain applications, especially those that require modeling longer sequences and more complex dependencies. The choice between GRU and LSTM often depends on the specific requirements of the task at hand. Despite its relative simplicity, GRU has proven to be a powerful tool in the deep learning toolkit.

3.2.5 BERT

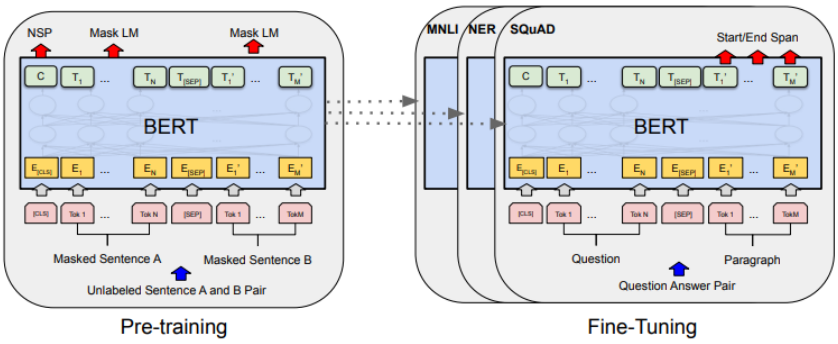

Fig. 5: Procedure of bert

One of the biggest challenges in natural language processing (NLP) is the shortage of training data. Because NLP is a diversified field with many distinct tasks, most task-specific datasets contain only a few thousand or a few hundred thousand human-labeled training examples. However, modern deep learning-based NLP models see benefits from much larger amounts of data, improving when trained on millions, or billions, of annotated training examples. To help close this gap in data, researchers have developed a variety of techniques for training general purpose language representation models using the enormous amount of unannotated text on the web (known as pre-training). The pre-trained model can then be fine-tuned on small-data NLP tasks like question answering and sentiment analysis, resulting in substantial accuracy improvements compared to training on these datasets from scratch. Open sourced a new technique for NLP pre-training called Bidirectional Encoder Representations from Transformers, or BERT. With this release, anyone in the world can train their own state-of-the-art question answering system (or a variety of other models) in about 30 minutes on a single Cloud TPU, or in a few hours using a single GPU. The release includes source code built on top of TensorFlow and a number of pre-trained language representation models. In our associated paper, we demonstrate state-of-the-art results on 11 NLP tasks, including the very competitive Stanford Question Answering Dataset (SQuAD v1.1). Input: The input data of BERT concludes 3 components which are

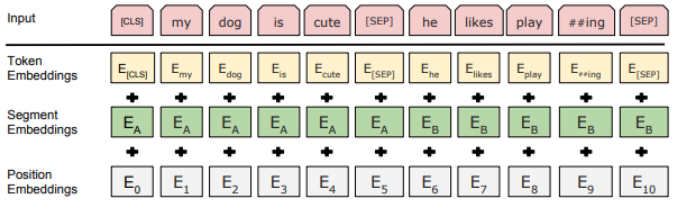

Fig. 6: Input of bert

token embedding which separates tokens in an input; segment embedding, this helps the model to distinguish different sentences in an input; finally positional embedding which indicate the position of each token.

MLM: The first idea used in BERT is Masked-Language Modeling (MLM). BERT tries to predict words in a sentence which are randomly masked before by it. The masking proportion is about 15 percent and the masked token will be replaced by token [MASKED]. BERT use the bidirectional approach, so that it can look both previous and next tokens, understand the full context of the sentence to predict the masked words.

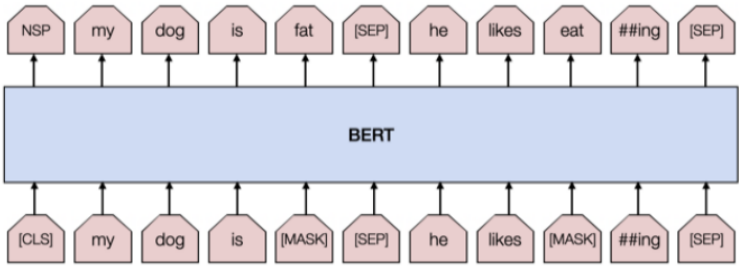

Fig. 7: MLM task

NSP: The second idea in BERT is Next Sentence Prediction (NSP). This technique is used to learn the relationship between two sentences. BERT receives the input of two sentences and tries to predict whether the second sentence is the next sentence of the first sentence. During training, half of time the truth second sentence is fed with the first sentence and half of time the second sentence is a random sentence.

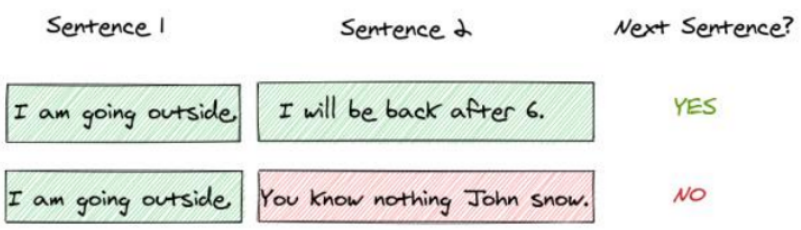

Fig. 8: NSP task

3.2.6 XLM-RoBERTa

XLM-RoBERTa model pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages. It was introduced in the paper Unsupervised Cross-lingual Representation Learning at Scale by Conneau et al.

XLM-RoBERTa is a multilingual version of RoBERTa. It is pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages.

RoBERTa is a transformers model pretrained on a large corpus in a self-supervised fashion. This means it was pretrained on the raw texts only, with no humans labelling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts.

More precisely, it was pretrained with the Masked language modeling (MLM) objective. Taking a sentence, the model randomly masks 15 percentage of the words in the input then run the entire masked sentence through the model and has to predict the masked words. This is different from traditional recurrent neural networks (RNNs) that usually see the words one after the other, or from autoregressive models like GPT which internally mask the future tokens. It allows the model to learn a bidirectional representation of the sentence. This way, the model learns an inner representation of 100 languages that can then be used to extract features useful for downstream tasks: if you have a dataset of labeled sentences for instance, you can train a standard classifier using the features produced by the XLM-RoBERTa model as inputs.

3.2.7 PhoBERT

PhoBERT, derived from RoBERTa, incorporates notable distinctions in both dataset selection and preprocessing techniques. Dat et.al identified two primary challenges in developing a Vietnamese language model.

Firstly, the absence of diverse monolingual training data, with only the Vietnamese Wikipedia corpus available for training, proved limiting. This dataset, employed in the pretraining of multilingual models like XLM-R, is constrained by its formality and relatively small size (1GB uncompressed). Expanding the pretraining data is crucial for enhancing pretrained language models.

Secondly, Vietnamese text employs white spaces to separate syllables within words. Traditional BERT-based models did not account for this linguistic nuance. The solution involved augmenting the pretraining data by incorporating a second corpus (19GB) sourced from news websites and topics. To address the whitespace issue, RDRSegmenter from VnCoreNLP was deployed for word and sentence segmentation, yielding 145 million word-segmented sentences. Subsequently, fastBPE, distinct from RoBERTa, was used to segment these sentences into subword units using a 64K subword type dictionary. Regarding optimization, PhoBERT utilized the RoBERTa implementation in fairseq, running for 40 epochs. For PhoBERT base, optimized by Adam with a batch size of 1024, a peak learning rate of 0.0004 was employed. PhoBERT large used a batch size of 512, a peak learning rate of 0.0002, and comprised 13 hidden states.

The parameters of PhoBERT were frozen, and the model's input traversed PhoBERT, with the output from some last hidden states fed into a subsequent layer. This subsequent layer involved a 2D convolutional neural network with m output channels, followed by 2D max pooling with a kernel size of 3. The resulting output underwent flattening, passing through a fully connected layer and a logsoftmax activation function.

4 Results

4.1 Metrics used

Let:

- TP_i (true possitive) is the number of instances t are assigned correctly to class c_i.
- TN_i true negative) is the number of instances inside c_i that are assigned correctly to another class.
- FP_i (false possitive) is the number of instances that are assigned incorrectly to classc_i.
- FN_i(false negative) is the number of instances inside c_i that are assigned incorrectly to another class.
- Percentage of correct predictions on testing data:

$$Accuracy = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}$$

- Percentage of correct instances, among all that are assigned to c_i:

$$Precision = \frac{TP_i}{TP_i + FP_i}$$

- Percentage of instances in c_i that are correctly assigned to c_i:

$$Recal(c_i) = \frac{TP_i}{TP_i + FN_i}$$

- F_1 - score is the harmonic mean of precision and recall. It can provide a unified view on the performance of a classifier and is computed as:

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

4.2 Model selection

To evaluate and choose the best hyperparameters for each model, we used Grid search with a 5–fold cross validation strategy. We then compare the cross validation's results by applying ttest for each pair. The models with tuned hyperparameters have been saved in the directory. Overall, all our deep earning models have some common training techniques as follow:

- Loss function cross entropy:

$$L_C E = -\sum_{i=1}^{n} t_i log(pi)$$

Where t_i is the truth label and p_i is the softmax probability for the i^th class.

- Weight initialization: Xavier initialization. All the weights of a layer L are picked randomly from a normal $\mu = 0$ and variance $\sigma^2 = \frac{1}{n^L}$ where $n^L$is the number of neurons in layer $L - 1$. This will prevent the gradients of network's activations from vanishing or exploding.
- Optimizer: Adam. It integrates the pros of both Momentum and RMSprop. It utilizes the squared gradients to scale the learning rate as RMSprop and it is similar to the momentum by using the moving average of the gradient. Its advantages are more memory efficient and less computational ower.
- Regularization technique: Dropout.

4.3 Results

Figure 4 shows the results of our models with the best embedding technique. Here, Pre stands for pretrained embedding:

Model	Accuracy	Precision	Recall	F1
Pre+RNN	48.99	56.89	48.99	51.22
TF-IDF+Random Forest	49.49	51	49	49
TF-IDF +XG-Boost	51.08	52	51	50
Word2Vec +GRU	51.37	53.00	51.37	50.93
Word2Vec +CNN	51.80	56.56	51.80	50.90
Word2Vec +LSTM	52.81	54.46	52.81	51.94
TF-IDF+SVM	53.54	55	54	53
XLM-roberta	65.2	65.1	65.2	64.9
PhoBERT	65.9	66.1	65.9	65.7

Fig. 9: Results of the models on our test dataset

Here are some observations we can have:

- SVM– despite being a simple model, has pretty good result with 53.54. We can see that TF – IDF with machine learning methods still perform quite well in the sentiment analysis task.
- CNN architecture's performance is slightly better than GRU in this case and just lower than LSTM in DNN architecture, but have training time much faster than LSTM.
- phoBERT, with huge improvements in dealing with Vietnamese language, has outperformed all other models with a big gap, outpacing the best Deep learning model we trained (pre + XLM-roberta+ Attention) by 3% in F1 score

Figure 5 shows the detailed classification result of phoBERT model on the benchmark dataset. We can observe that the results are quite good with 0.66 and 0.657 F1 – score, respectively.

	precision	recall	f1-score	support
0	0.500	0.400	0.444	40
1	0.601	0.720	0.655	132
2	0.689	0.756	0.721	193
3	0.756	0.674	0.713	46
4	0.593	0.543	0.567	129
5	0.765	0.672	0.716	116
6	0.700	0.568	0.627	37
accuracy			0.659	693
macro avg	0.658	0.619	0.635	693
weighted avg	0.661	0.659	0.657	693

Fig. 10: Classification result of phoBERT on benchmark dataset

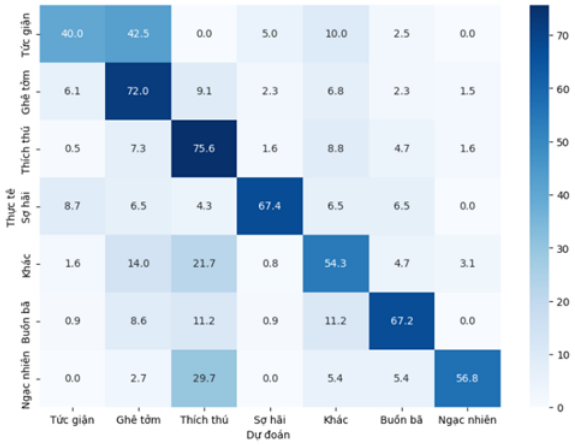

Fig. 11: Confusion matrix of the best classification model on the UIT-VSMEC

5 Conclusion and future work

In this study, we use a human-annotated corpus for emotion recognition for Vietnamese social media text for research purpose and achieved 6,927 sentences annotated with one of the seven emotion labels namely enjoyment, sadness, anger, surprise, fear, disgust and other with the annotation agreement of over 82%. We also presented machine learning and deep neural network models used for classifying emotions of Vietnamese social media text. In addition, we reached the best overall weighted F1-score of 66% on the original UIT-VSMEC corpus with Phobert. In the future, we want to improve the quantity as well as the quality of the corpus due to its limitation of comments expressing emotions of anger, fear and surprise. Besides, we aim to conduct experiments using other machine learning models with distinctive features as well as deep learning models with various word representations or combine both methods on this corpus.

6 References

[1] NGUYEN, Dat Quoc; NGUYEN, Anh Tuan. PhoBERT: Pre-trained language models for Vietnamese. arXiv preprint arXiv:2003.00744, 2020.

[2] DEVLIN, Jacob, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[3] DANG, Nhan Cach; MORENOGARCÍA, María N.; DE LA PRIETA, Fernando. Sentiment analysis based on deep learning: A comparative study. Electronics, 2020, 9.3: 483.

[4] LIU, Yinhan, et al. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

[5] CUI, Zhiyong, et al. Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. arXiv preprint arXiv:1801.02143, 2018.

[6] BAHDANAU, Dzmitry; CHO, Kyunghyun; BENGIO, Yoshua. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

[7] VASWANI, Ashish, et al. Attention is all you need. Advances in neural information processing systems, 2017, 30.

[8] NGUYEN, Khang Phuoc-Quy; VAN NGUYEN, Kiet. Exploiting vietnamese social media characteristics for textual emotion recognition in vietnamese. In: 2020 International Conference on Asian Language Processing (IALP). IEEE, 2020. p. 276-281.

[9] 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.

[10] 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.

[11] HOANG, Suong N., et al. An efficient model for sentiment analysis of electronic product reviews in Vietnamese. In: International Conference on Future Data and Security Engineering. Springer, Cham, 2019. p. 132-142.

[12] ZHANG, Ye; WALLACE, Byron. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820, 2015.

[13] RAFFEL, Colin, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 2020, 21.140: 1-67.

[14] SUN, Chi, et al. How to fine-tune bert for text classification?. In: China national conference on Chinese computational linguistics. Springer, Cham, 2019. p. 194- 206.

[15] Vong Anh Ho, et al. Emotion Recognition for Vietnamese Social Media Text. In Proceeding of PACLING, 2019