Hate Speech Detection on Vietnamese Social Media Texts using the PhoBERT-CNN Model

Khanh Quoc Tran[1,2], Huy Hoang Nguyen[1,2], An Tran-Hoai Le[1,2], and Hop Trong Do[1,2]
18520908@gm.uit.edu.vn, 18520842@gm.uit.edu.vn, 18520426@gm.uit.edu.vn, hopdt@uit.edu.vn

[1] University of Information Technology, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam

Abstract. In this paper, inspired by researches on Hate Speech Detection, we decide to continue enhancing the performance of the social comments classification task from the ViHSD dataset and HSD-VLSP dataset. To solve this, firstly, the given datasets must go through a nine-step preprocessing, which is divided into 2 phases, to get the ideal data. Secondly, we conduct three states of the arts approaches for both preprocessed datasets, which are Deep Learning approach with Text-CNN, BiLSMT with a variant of word embeddings; Transfer Learning approach with XLMR, RoBERTa, BERT cased and PhoBERT; Combination approach that is combining two best models from previous approaches. After the results for each model, we achieve the best performance for both datasets when we use the combination model with F1-score of 64,43% for the ViHSD dataset and 90.89% for the HSD-VLSP dataset.

Keywords: Hate speech detection · Text classification · Social media texts · Combine model · Comparison · Big Data · Spark NLP · PyTorch

1 Introduction

In modern life, social networks are a must to many individuals, especially in Covid-19 pandemics. It helps to connect people, sharing their likes and opinions. However, many of us sit in front of the screen and exploit this platform and their freedom of speech power to harm others by their thoughtless language. Unfortunately, research did in 2017 [21] convinced that toxic speech could have a tremendous effect on our health. Because of this, the task of hate speech detection is still a hot issue, not only in English but also in Vietnamese. When this task is solved, it will make a significant impact on user experience on online platforms.

Knowing this urge, we have two contributions to the Vietnamese hate speech detection task in ViHSD of [20] and HSD-VSLP of [29] dataset in this paper. Firstly, we propose two phases of data preprocessing that focus on normalizing tricky comments and posts from the given datasets into having the same standards before getting rid of unnecessary components. Secondly, we conduct abundant

experience by using novelty and state-of-the-art approaches for both datasets and creatively combining the bests to introduce a simple but effective combination model for this task.

We decided to implement some models built on Spark NLP to compare with our proposed method. Spark NLP is an open-source natural language processing library, built on top of Apache Spark and Spark ML. It provides an easy API to integrate with ML Pipelines and it is commercially supported by John Snow Labs [15]. Spark NLP's annotators utilize rule-based algorithms, machine learning and some of them Tensorflow running under the hood to power specific deep learning implementations. The library covers many common NLP tasks, including tokenization, stemming, lemmatization, part of speech tagging, sentiment analysis, spell checking, named entity recognition, and more. The full list of annotators, pipelines, and concepts is described in the online reference[3].

2 Related Works

Hate speech detection task in recent years is the populous topic of Natural Language Processing. This task also is adapted to many Vietnamese by creating redundant datasets with enormous approaches are done. As a result, in order to have an overview of this task, we divide it into English hate speech detection researches and Vietnamese hate speech detection researches.

Hate speech detection in English is a well-developed task. For instance, Automated Hate Speech Detection and the Problem of Offensive Langues by Davidson et al. [4] uses Logistic Regression and achieved 90% in F1-score or Deep Learning for Hate Speech Detection in Tweets [1] also achieved 93% in F1-score with LSTM + Random Embedding + GBDT.

Not like in English, the hate speech detection task in Vietnamese is new. Consequently, not much in performance is achieved, but those efforts from researchers will build a firm foundation for this task. Constructive and Toxic Speech Detection for Open-domain Social Media Comments in Vietnamese by Nguyen et al. [23] with the best performance at 59.4% in F1-score; Emotion Recognition for Vietnamese Social Media Text by Ho et al. [8] with 59.74% in F1-score when using CNN model.

3 Dataset

3.1 Task Definition

In this section, we summarize the Vietnamese Hate Speech Detection Task [20]. This task aims to detect whether a comment on social media is hate or offensive, or clean. Formally, the task is described as follows.

Input: Given Vietnamese comments on the social networks site.

Output: One of three different labels is predicted by classifiers.

[3] Spark NLP Quickstart - `https://nlp.johnsnowlabs.com/docs/en/quickstart`

• **Hate speech (HATE)** contains abusive language, which often bears the purpose of insulting individuals or groups and can include hate speech, derogatory and offensive language. An item is identified as hate speech if it (1) targets individuals or groups based on their characteristics; (2) demonstrates a clear intention to incite harm or to promote hatred; (3) may or may not use offensive or profane words.

• **Offensive but not hate speech (OFFENSIVE)** is an item (posts/ comments) that may contain offensive words, but it does not target individuals or groups based on their characteristics.

• **Neither offensive nor hate speech (CLEAN)** is a normal item. It is conversations, expressing emotions normally. It does not contain offensive language or hate speech.

3.2 Dataset

We use the Vietnamese Hate Speech Detection datasets, including ViHSD (Luu et al., 2021) [20] and HSD-VLSP (Vu et al., 2019) [29] for our experiment. ViHSD dataset [20] is consists of 33,400 comments on social media. Four independent annotators annotate each comment on this dataset with an inter-annotator agreement score of Cohen Kappa at $K = 0.52$. This dataset is divided into training: dev: test sets with a 70: 10: 20 rate. The HSD-VLSP dataset is a Vietnamese comments dataset about Hate Speech Detection on social networks provided by the VSLP 2019 shared- task (Vu et al., 2019) [29]. This dataset contains the comments and posts on Facebook social networks, including 25,431 items. Each data line of the training dataset is assigned one of three labels CLEAN, OFFENSIVE, or HATE. There is a huge difference in the number of comments labeled CLEAN compared to comments labeled OFFENSIVE and HATE. Besides, We found that comments are often short comments because users tend to be brief and use many acronyms. Table 1 and Figure 1 shows overview statistics of the dataset.

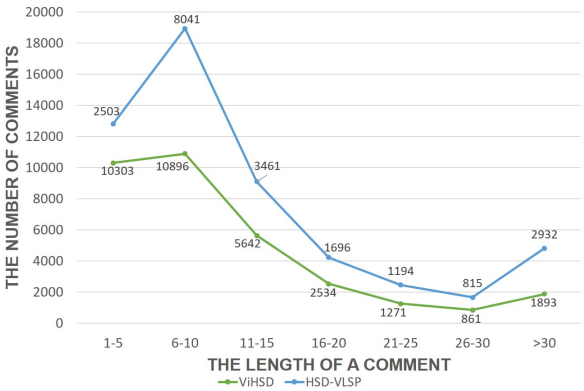


Fig. 1: Distribution of the comments length in the two datasets.

Table 1: Overview statistics of the two Vietnamese hate speech detection datasets.

Datasets	Labels	Percentage (%)	Example
ViHSD	CLEAN	82.71	Link đâu thằng kia (**English:** Where is the link, man)
	OFFENSIVE	6.77	vkl. (**English:** Cuss.)
	HATE	10.52	Thầy đ*t mẹ giả tạo vl =)) (**English:** The teacher is so f*cking fake =))
HSD-VLSP	CLEAN	91.49	Cho xíu nhạc đi (**English:** Some music please.)
	OFFENSIVE	5.02	đ*o lấy vk nữa đâu (**English:** no more f*cking married)
	HATE	3.49	Thằng già ch* ch*t (**English:** F*ck that old man)

4 Our Approach

4.1 Proposed System

In this section, we propose our simple and efficient approach for this task. We only focus on the combined PhoBERT-CNN model for generating a best-performance model by fine-tuning techniques. Figure 2 shows the overview of the approach using two essential components, which are pre-processing techniques (see Subsection 4.2), and the core method PhoBERT-CNN (see Subsection 4.3).

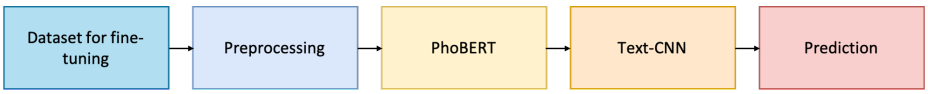


Fig. 2: Our propose approach for hate speech detection of Vietnamese comments.

4.2 Data Preprocessing

The data preprocessing step is a vital step in most Machine Learning or Deep Learning projects. Because of that, in order to clean the given dataset well, we propose a two-phase data preprocess, and by doing this, this paper is a novelty compared to baselines. The Figure 3 below again makes an overview of Data Preprocessing process.

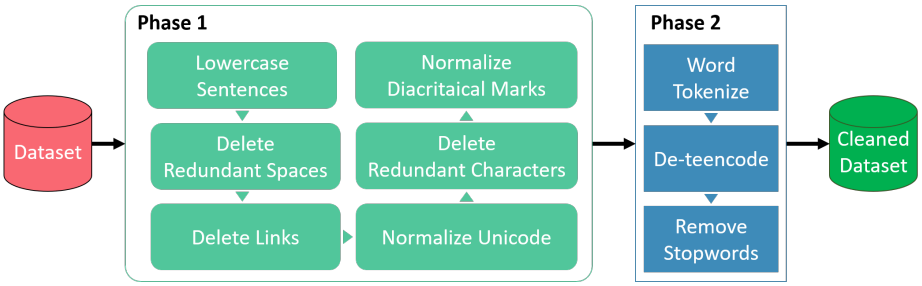


Fig. 3: Data preparation steps.

4.2.1 Phase 1:

• **Lowercase Sentences**: We lowercase all the characters of all the data points in the datasets. We do this to avoid Python sees two exact words as separate because of their forms.

• **Delete Redundant Space**: Users on social media unwittingly or wittingly type multiple spaces on their comments. As a result, we decide to remove redundant spaces.

• **Delete Links**: We believe that website links that are inside a comment do not affect the sentiment of the comment. As a result, we decide to get rid of them from all the comments.

• **Normalize Unicode**: We also see a lot of Vietnamese words in the dataset that are the same, but python sees them as separate because of its Unicode. The reason why is that there are many Unicode Transformation Formats (UTF) such as UTF-8, UTF-16, UTF-32, and so on are used widely, but our choice is normalizing to UTF-8.

• **Delete Redundant Characters**: We remove redundant characters that the users intentionally make.

Fig. 4: An example of a comment which generally means "If have nothing to say, have a call to see each other is happy enough. So f*cking simple" and deleted redundant characters version of it.

Figure 4 above is an instance for deleting redundant characters process. As we can see, the word "vuiiiii", which means happy, and the word "vllll", which is an offensive teen code in Vietnamese but does not have a clear meaning in English, after cut off the redundant characters, they will become "vui" and "vl". However, the word "kk" is also a teen code for the action of laughing - kaka, and

the word "call" is an English word are not removed duplicate characters "k" and "l" due to theirs valuable meanings and if the word "kk" becomes "k", it will be a different teen code that means no in English.

• **Normalize Accented Letters**: Because of the mix of diacritical marks style, we decide to normalize diacritical marks of data by followings rules:

– If there is one vowel, the diacritical marks will be on that vowel. For example: má (mom), lá (leaf), mê (like).
– If there are two vowels, the diacritical marks will be on the first one, for example: lóa (shinny), qùa (gift). If three vowels or two vowels follow with a consonant, the diacritical will be on the second vowel, for example: khuỷu (elbow), quán (store).
– "ê" and "ơ" are exceptional because diacritical marks always are on them, for example: khuyển (dog), quở (reproachfully).

Table 2: Number of changes on the datasets after the Phase 1.

Datasets	Lowercase	Redundant		Inconsistent		Link
		Spaces	Characters	Unicode	Accented words	
ViHSD	28,540	488	2,127	753	620	21
HSD-VLSP	0	1	2,667	0	761	1

All the above steps in Phase 1 are conducted in the same order as the listing. The output of this Phase 1 is fed directly to the next Phase 2.

4.2.2 Phase 2:

• **Word tokenzie**: The input sentence is splitting into words or meaningful word phrases. In order to do this, we use Word Segmenter of VnCoreNLP [28] for the PhoBERT model and NLTK [2] for other models.

• **De − teencode**: In the online space, people usually have a significant amount of their time chit chat and they also often use the short form of words to type faster, some are used to trick the systems when they are swearing. Moreover, those abbreviations also have their name in Vietnam, teen codes. As a result, to help our models better understand the sentences, we have to map those teen codes into ordinary words. Furthermore, that process of mapping teen codes, we named it De–teen code, and the following Table 3 shows some instances of them.

Table 3: Examples of teencodes and their expansions.

No.	Teencode	De-Teencode	
		Vietnamese sentences	English meanings
1	đc đấy	được đấy	nice
2	ko	không	no
3	cc	con c*c	d*ck

• **Remove stopwords**: We also remove stopwords from the comments because of their meaninglessness. In our experiments, we use the Vietnamese stopword dictionary [16] for removing stop words in the sentence.

In this Phase 2, the data are tokenized, de-teencodes, and removed stopwords. Phase 2 has that order because the output of Word tokenizers is a list of words, word phrases, and characters that separate with others by space. Those characters then are checked if they are teencode and will be De–teencoded in the next step. Therefore, De–teencode step follows after the Word tokenize step is a wise decision. Finally, after De–teencode step, we remove all stopwords, and the reason why we remove stopwords after De – teen code is that those teencodes possibly are stopwords also.

Table 4: The number of teencodes and stopwords of the datasets.

Datasets	**Teencodes**		**Stopwords**		**#Words**
	Frequency	**Percentage**	**Frequency**	**Percentage**	
ViHSD	15,344	4.00%	153,330	40,01%	383,270
HSD-VLSP	13,757	3.24%	127,531	30.01%	424,301

4.3 PhoBERT-CNN

We conduct this approach because combining PhoBERT and Text-CNN has the best performance among transfer learning models and deep learning models, respectively. PhoBERT has duty on extracting features from sentences for input of Text-CNN classification model. Figure 5 presents the architecture of our approach for Vietnamese hate speech detection (Vietnamese HSD).

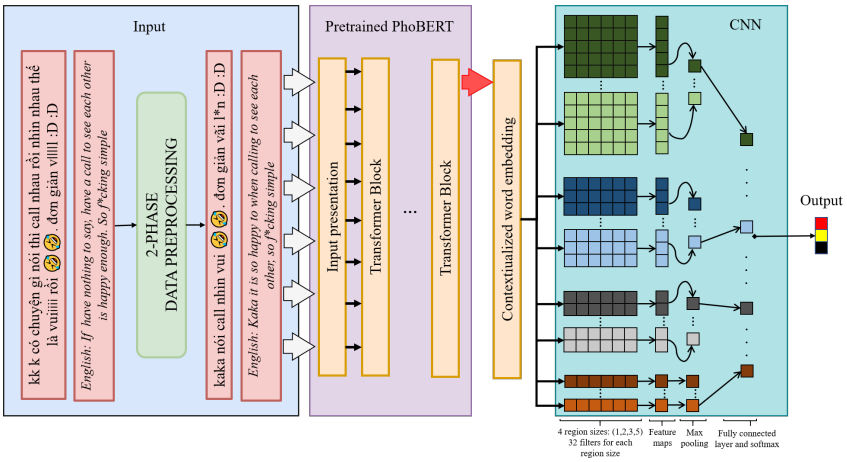


Fig. 5: An overview of our Vietnamese HSD system using PhoBERT-CNN.

5 Experiments and Results

5.1 Evaluation Metric

Before going through experimental results, we first discuss the evaluation metrics used in this paper. In Luu et al. study [20], he uses the Accuracy and F1–macro score in order to evaluate his models. Therefore, we also decided to use the Accuracy score and F1-macro score (%) to measure the performance of models.

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{1}$$

$$Precision = \frac{\text{True Positive}}{\text{True Positive + False Positive}} \tag{2}$$

$$Recall = \frac{\text{True Positive}}{\text{True Positive + False Negative}} \tag{3}$$

$$F1 = 2 * \frac{\text{Precision * Recall}}{\text{Precision + Recall}} \tag{4}$$

Because the given datasets have significantly imbalanced classes, the F1-macro score, which weights accuracy and recall equally for each label before finding their unweighted mean, is the best measurement for this task. As a result, we pay much effort into making this score higher rather than accuracy.

5.2 Baseline System

5.2.1 Multinomial Naive Bayes: This is an algorithm based on the Bayes theorem of probability theory to make predictions and classify data based on observed data and statistics [26,14].

Multinomial Naive Bayes (MultinomialNB) Classification is one of the many algorithms used in machine learning to make the most accurate predictions on a collected dataset because it is relatively easy to train and achieve high performances. It belongs to the supervised machine learning group, which means machine learning model trained on annotated data samples.

5.2.2 Logistic Regression: This is one of the basic and well-known methods of classification algorithms, especially binary classification. Logistic regression has become an important tool in the discipline of machine learning. The approach allows an algorithm being used in a machine learning application to classify incoming data based on historical data.

A Logistic Regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables. For example, a logistic regression can be used to predict whether any comment on a social network contains negative, offensive content to specific audiences. In text classification, it requires manual features extracted from data [7,9].

5.2.3 Decision Tree: Decision tree is the most powerful and popular tool for classification, and prediction [25,12]. A Decision tree is a flowchart like a tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) stores a class label.

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from preliminary data(training data).

In Decision Trees, we start from the root of the tree for predicting a class label for a record. We compare the values of the root attribute with the record's attribute. Based on the comparison, we follow the branch corresponding to that value and jump to the next node.

5.2.4 Random Forest: Random Forest is a machine learning algorithm built on multiple sets of Decision Tree. The model's output is based on the aggregate decision on the decision trees it generates with the voting method. Random Forest is a Supervised Learning method to handle classification and regression problems.

Random Forest gives us a very accurate result with such a mechanism, but the trade-off is that we cannot understand how this algorithm works due to the complicated structure of this model. This is one of the Black Box methods - that is, we put our hands inside and get the results, but cannot explain the mechanism of the model [17,13].

5.2.5 Transfer Learning - PhoBERT: The transfer learning model has attracted increasing attention from NLP researchers around the world for its outstanding performances. One of the SOTA language models as BERT, which stands for Bidirectional Encoder representations from transformers, is published Devlin et al. [5]. It is a bi-directional transformer model for pre-training over lots of text data with no label to understand a language representation. Then, we fine-tune for specific problems.

For Vietnamese, the SOTA method was first released and called PhoBERT by Nguyen et al. [22] for solving Vietnamese NLP problems. PhoBERT is a pre-trained model, which has the same idea as RoBERTa, a replication study of BERT is released by Liu et al. [18], and there are modifications to suit Vietnamese.

5.3 Experimental Settings

As we described in section 4, we experiment on the given datasets with three approaches: the Machine Learning, the Transfer Learning, and the Combine approach.

5.3.1 Machine Learning approach: This paper uses these Machine Learning models: ial Naive Bayes, Logistic Regression, Decision Tree, and Random

Forest. Beside, we have used these models with TF-IDF technique through TfidfVectorizer function in PySpark[4] library with parameter ngram_range is (1,2). The class-weight algorithm is used to deal with the imbalance in the datasets, but we do not get any better results.

• **Multinomial Naive Bayes:** We use MutinomialNB with "alpha" = 1.0.

• **Logistic Regression:** Set the Logistic Regression model parameters with C = 1.0, solver = "lbfgs", maxIter = 20, and regParam = 0.3.

• **Decision Tree:** This model is run with n_estimators = 108, random_state = "None", class_weight = "balanced", max_depth = 17, and min_samples_leaf = 3.

• **Random Forest:** We implement a Random Forest Classifier model with numTrees = 200, maxDepth = 10, maxBins = 64.

5.3.2 Tranfer Learning approach: We implement PhoBERT based [22] for this approach. It runs with three epochs, the max sequence length is 60, the batch size is 64, the learning rate is at 3e-5, accumulation steps are 5, learning rate decay steps are 70. We also experiment on this dataset with the XLM-R [3], RoBERTa [18], and BERT cased [5] models but do not have better performance than proposed approach.

5.3.3 Our approach (PhoBERT-CNN): In this approach, we combine the PhoBERT based pre-trained model [22] from HuggingFace with the Text-CNN model. The output of PhoBERT based pre-trained is used as input embedding for Text-CNN.

• PhoBERT based pre-trained is initialized with a max sequence length is 20.

• Text-CNN is built with four layers of conv1D with filter size is 32 and size 1, 2, 3, 5, respectively.

This combined model has an Adam optimizer, the learning rate is 2e-5, epsilon is 1e-8, and dropout is 0.4. This combined model has an Adam optimizer, the learning rate is 2e-5, epsilon is 1e-8, and dropout is 0.4.

5.4 Experimental Results

Our experiments achieve improved results, creating dictionaries that enhance our single model's performance 2%. With our single models, PhoBERT archives the best results when making predictions on the ViHSD [20] and HSD-VLSP [29] datasets. PhoBERT can perform parallel computations for words, reduce vanishing gradients, and helping the model learn better. With a combination of the single models' strengths, our combining PhoBERT-CNN model has achieved outperformance results. Table 5 shows our results through experiments performed.

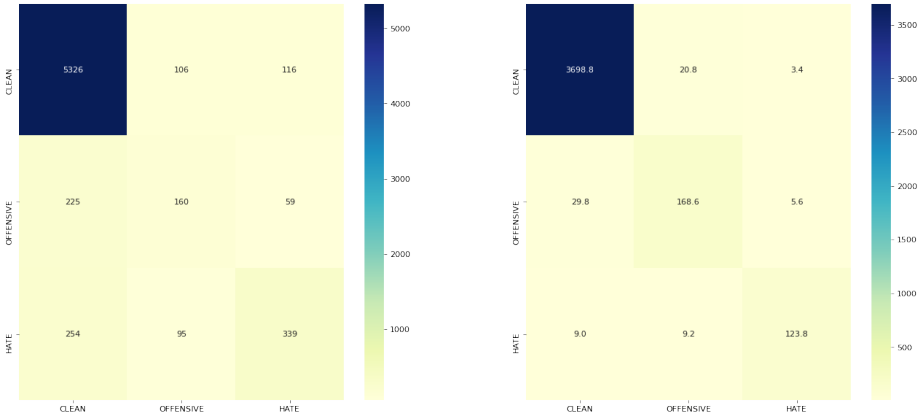
[4] `https://spark.apache.org/docs/latest/mllib-feature-extraction.html#tf-idf`

Table 5: Evaluation results on the two Vietnamese hate speech detection datasets.

Models	ViHSD		HSD-VLSP	
	F1-score	Accuracy	F1-score	Accuracy
Multinomial Naive Bayes	61.67	86.98	85.60	97.14
Logistic Regression	62.46	86.78	86.23	97.22
Desion Tree	60.49	85.89	84.52	95.52
Random Forest	63.51	87.13	85.36	96.11
ClassifierDL + UniversalSentenceEncoder	62.15	86.72	84.90	95.75
ClassifierDL + BERT	63.68	86.33	85.69	96.89
ClassifierDL + Glove	63.93	87.08	85.40	96.37
XLM-R	62.38	83.62	86.57	97.15
RoBERTa	61.49	83.05	85.79	96.95
BERT	60.29	84.52	85.41	96.19
PhoBERT	63.01	86.11	86.68	97.58
Our approach (PhoBERT-CNN)	**64.43**	**87.17**	**90.89**	**98.26**

5.5 Error Analysis

Error analysis is carried out to analyze the errors that we encountered in our system by quantitative analysis using the confusion matrix of our best-performing model. Figure 6 shows the confusion matrix of our best model when predicting hate speech detection on the test set. We see that our system's ability to predict our system on the CLEAN label is better than the OFFENSIVE label and HATE label.



(a) ViHSD dataset. (b) HSD-VLSP dataset.

Fig. 6: Confusion matrix of our propose for Vietnamese hate speech detection.

There are still some opinions of misclassification in the data set due to the ambiguity in identifying the labels. We can see that many misclassified comments are affected by the decision keywords, such as in the label CLEAN but are misclassified in the topic HATE due to the keyword "lon", "vl", "đm". In addition, the HATE label and OFFENSIVE label is often confused with the CLEAN label because it contains racist or insinuating content that makes it difficult to predict.

Table 6: Several examples of classification error on the given datasets.

Title	**Label**	**Predict**
Đừng cố biện minh =)))) choi lon (**English:** Don't try to make excuses, play big)	CLEAN	HATE
Lắm tiền mà chả có nổi ý thức của loài người :)) (**English:**Money can't buy human consciousness :)))	CLEAN	HATE
con này hết thuốc chữa rồi (**English:** I'm done with this dumb ass)	HATE	CLEAN
Nham (**English:** Bull sh*t)	OFFENSIVE	HATE

5.6 Comparison With Previous Studies

The result of our approach is better than those of previous studies [20] conducted on the same dataset. We follow the same metrics for evaluating, macro F1-score (%) for HSD-VLSP [29] and macro F1-score and Accuracy (%) for ViHSD [20]. Besides, because of the privacy of HSD-VLSP, we use KFoldCrossvalidation (k=5) for training our models as a solution of lacking test set. Table 7 and 8 shows the best results we achieved compared to previous studies.

Table 7: The comparison with previous studies on ViHSD dataset.

Model	**F1-score**	**Accuracy**
Text-CNN + fastText (Luu et al., 2021) [20]	61.11	86.69
GRU + fastText (Luu et al., 2021) [20]	60.47	85.41
BERT (Luu et al., 2021) [20]	**62.69**	**86.88**
XLM-R (Luu et al., 2021) [20]	61.28	86.12
DistilBERT (Luu et al., 2021) [20]	62.42	86.22
Our approach (PhoBERT-CNN)	**64,43**	**87,17**

Table 8: The comparison with previous studies on HSD-VLSP dataset.

Model	**F1-score**
Text-CNN (Luu et al., 2020) [19]	83.04
Logistic regression* (Pham et al., 2019) [10]	61.97
Logistic regression + Random Forest + Extra Tree* (Dang et al., 2019) [27]	58.88
DCNN, Text-CNN, LSTM, LSTMCNN, SARNN* (Nguyen et al., 2019) [24]	58.45
BiLSTM (Do et al., 2019)* [6]	56.28
CNN + BiLSTM + LSTM (Huynh et al., 2020) [11]	**86.96**
Our approach (PhoBERT-CNN)	**90.89**

*Indicates that the result is evaluated on a test set of the VLSP shared task 2019. Others use k-fold cross-validation to evaluate the model (k=5) following the study.

6 Spark Streaming

In today's technology-driven world, every second, a vast amount of data is generated. Constant monitoring and proper analysis of such data are necessary to draw meaningful and valuable insights.

As Data scientists, in the era of continuous data generation, we need to process data in real-time. A traditional data processing system such as the Extract Transform Load (ETL) process is only adequate for processing static data. Traditional ETL tools process data in batches and cannot handle real-time data, but only data that is already stored in a database system.

Real-time data from sensors, IoT devices, log files, social networks, etc., need to be closely monitored and immediately processed. Therefore, we need a highly scalable, reliable, and fault-tolerant data streaming engine for real-time data analytics.

6.1 Data Streaming

Data streaming is a way of collecting data continuously in real-time from multiple data sources in the form of data streams. Datastream can be thought of as a table that is continuously being appended.

Data streaming is essential for handling massive amounts of live data. Such data can be from various sources like online transactions, log files, sensors, in-game player activities, etc.

We need systems that support stream processing for the processing of real-time data, such as Apache Spark. Those processing systems provide the option for continuous computations, as data is continuously flowing through them. Examples of such functionalities are the regular cleaning and aggregation of the incoming data before storage.

There are various real-time data streaming techniques like Apache Kafka, Spark Streaming, Apache Flume, etc. In this paper, we will implement data streaming using Spark Streaming.

6.2 Our goal

This project is a good starting point for those who have little or no experience with Apache Spark Streaming. We use Twitter data since Twitter provides an API for developers that is easy to access. We present an end-to-end architecture on how to stream data from Twitter, clean it, and apply a combined PhoBERT-CNN model to detect the hate or offensive of each tweet.

Input data: Live tweets with a keyword.
Main model: Data preprocessing and hate speech detection on the tweets.
Output: A parquet file with all the tweets and their hate speech prediction.

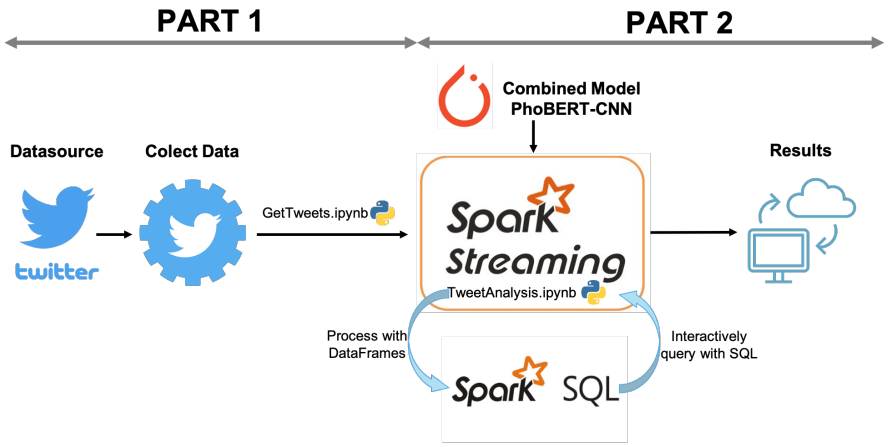


Fig. 7: The end-to-end architecture of our system using Spark Streaming.

6.3 Instructions

6.3.1 Part 1 - Send tweets from the Twitter API: In this part, we use our developer credentials to authenticate and connect to the Twitter API. We also create a TCP socket between Twitter's API and Spark, which waits for the call of the Spark Structured Streaming and then sends the Twitter data. Here, we use Python's Tweepy library for connecting and getting the tweets from the Twitter API.

6.3.2 Part 2- Tweet preprocessing and sentiment analysis: In this part, we receive the data from the TCP socket and preprocess it with the pyspark library, which is Python's API for Spark. Then, we apply hate speech detection on Vietnamese social media texts using the PhoBERT-CNN model, which is our proposed effective solution for this task. After hate speech sentiment analysis, we save the tweets and the analysis scores in a parquet file, which is a data storage format. These results will be used as a recommendation for the system administrator to decide on whether to remove hate and offensive tweets or not.

6.4 Advantages of Spark Streaming

- Unified streaming framework for all data processing tasks(including machine learning, graph processing, SQL operations) on live data streams.
- Dynamic load balancing and better resource management by efficiently balancing the workload across the workers and launching the task in parallel.
- Deeply integrated with advanced processing libraries like Spark SQL, MLlib, GraphX.
- Faster recovery from failures by re-launching the failed tasks in parallel on other free nodes.

7 Conclusion and Future works

This paper has made two significant contributions, introduced a new data processing process with two phases to clean the given dataset well, and proposed a unique but effective solution for Vietnamese hate speech detection based on combined model PhoBERT-CNN. The combined model that we proposed gives better results than the single models and previous studies on the same data set and evaluation measure. We achieved F1-score results on the ViHSD and HSD-VLSP datasets of 64.43% and 90.89%, respectively. Furthermore, we have successfully built a real-time hate speech detection system for Vietnamese using Spark streaming. The results obtained are pretty optimistic and are a reliable basis for solving the task, helping to reduce the occurrence of hate or offensive comments, contributing to building an increasingly healthy online environment and safety.

In the future, we plan to implement some monolingual models for Vietnamese and setting experiments with various parameters for the approach we proposed to find a better model, improves the quality of the dataset, and have better performance in the Hate Speech Detection task.

References

1. Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on World Wide Web companion*, pages 759–760, 2017.
2. Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, 2006.
3. Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.
4. Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, 2017.
5. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

6. Hang Thi-Thuy Do, Huy Duc Huynh, Kiet Van Nguyen, Ngan Luu-Thuy Nguyen, and Anh Gia-Tuan Nguyen. Hate speech detection on vietnamese social media text using the bidirectional-lstm model. *arXiv preprint arXiv:1911.03648*, 2019.
7. Alexander Genkin, David D Lewis, and David Madigan. Large-scale bayesian logistic regression for text categorization. *technometrics*, 49(3):291–304, 2007.
8. Vong Anh Ho, Duong Huynh-Cong Nguyen, Danh Hoang Nguyen, Linh Thi-Van Pham, Duc-Vu Nguyen, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. Emotion recognition for vietnamese social media text. In *International Conference of the Pacific Association for Computational Linguistics*, pages 319–333. Springer, 2019.
9. David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.
10. Quang Pham Huu, Son Nguyen Trung, and Hoang Anh Pham. Automated hate speech detection on vietnamese social networks. Technical report, EasyChair, 2019.
11. Huy Duc Huynh, Hang Thi-Thuy Do, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. A simple and efficient ensemble classifier combining multiple neural network models on social media datasets in vietnamese. *arXiv preprint arXiv:2009.13060*, 2020.
12. M Ikonomakis, Sotiris Kotsiantis, and V Tampakas. Text classification using machine learning techniques. *WSEAS transactions on computers*, 4(8):966–974, 2005.
13. Md Zahidul Islam, Jixue Liu, Jiuyong Li, Lin Liu, and Wei Kang. A semantics aware random forest for text classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1061–1070, 2019.
14. Sang-Bum Kim, Hae-Chang Rim, DongSuk Yook, and Heui-Seok Lim. Effective methods for improving naive bayes text classifiers. In *Pacific rim international conference on artificial intelligence*, pages 414–423. Springer, 2002.
15. Veysel Kocaman and David Talby. Spark nlp: Natural language understanding at scale. *Software Impacts*, 8:100058, 2021.
16. Van-Duyet Le. stopwords: Vietnamese. https://github.com/stopwords/vietnamese-stopwords, 2017.
17. Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.
18. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
19. Son T Luu, Hung P Nguyen, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. Comparison between traditional machine learning models and neural network models for vietnamese hate speech detection. In *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 1–6. IEEE, 2020.
20. Son T Luu, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. A large-scale dataset for hate speech detection on vietnamese social media texts. *arXiv preprint arXiv:2103.11528*, 2021.
21. Shruthi Mohan, Apala Guha, Michael Harris, Fred Popowich, Ashley Schuster, and Chris Priebe. The impact of toxic language on the health of reddit communities. In *Canadian Conference on Artificial Intelligence*, pages 51–56. Springer, 2017.
22. Dat Quoc Nguyen and Anh Tuan Nguyen. Phobert: Pre-trained language models for vietnamese. *arXiv preprint arXiv:2003.00744*, 2020.
23. Luan Thanh Nguyen, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. Constructive and toxic speech detection for open-domain social media comments in vietnamese. *arXiv preprint arXiv:2103.10069*, 2021.

24. Thai Binh Nguyen, Quang Minh Nguyen, Thu Hien Nguyen, Ngoc Phuong Pham, The Loc Nguyen, and Quoc Truong Do. Vais hate speech detection system: A deep learning based approach for system combination. *arXiv preprint arXiv:1910.05608*, 2019.
25. Tomas Pranckevičius and Virginijus Marcinkevičius. Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal of Modern Computing*, 5(2):221, 2017.
26. Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.
27. Dang Van Thin, Lac Si Le, and Ngan Luu-Thuy Nguyen. Nlp@ uit: Exploring feature engineer and ensemble model for hate speech detection at vlsp 2019. *TRAINING*, 5:3–51, 1991.
28. Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. Vncorenlp: a vietnamese natural language processing toolkit. *arXiv preprint arXiv:1801.01331*, 2018.
29. Xuan-Son Vu, Thanh Vu, Mai-Vu Tran, Thanh Le-Cong, and Huyen Nguyen. Hsd shared task in vlsp campaign 2019: Hate speech detection for social good. *arXiv preprint arXiv:2007.06493*, 2020.