Vietnamese Hate Speech Detection in Streaming data with BigDL and Spark

Huy Hoang Nguyen[1,2,3], Dinh Tu Vo[1,2,3], and Trong-Hop Do[1,2,4]

[1] University of Information Technology, Ho Chi Minh City, Vietnam
[2] Vietnam National University Ho Chi Minh City, Vietnam
[3] {18520842,18521589}@gm.uit.edu.vn
[4] Correspondence: hopdt@uit.edu.vn

Abstract. This paper proposes an real-time hate speech detection system for streaming social media text. Many deep learning techniques is proposed for social media hate speech detection. The experiment was conducted using the VLSP-HSD dataset. BigDL framework is used to set up deep learning models while Spark is used for streaming data. A real-time hate speech detection system that detects and processes online hate speech in real-time was built to consolidate the effectiveness of the proposed system.

Keywords: hate speech detection · social network text · natural language processing · deep learning · streaming data.

1 Introduction

Social media platforms such as Facebook, TikTok and YouTube have been growing in popularity in recent years. There has been an increase in social media usage, particularly during the Covid-19 pandemic [7]. This resulted in a large amount of information, including hate speech. Many well-known platforms are demonstrating their efforts to address this issue through policies and technologies.

The difficulty is that it is hard to judge if the content is a violation or not without all the background information, such as other connected content or articles. That burning issue is even more difficult in the case of Vietnamese due to our language's varied vocabulary and complicated syntax, as well as the fact that most social media users utilize abbreviations and idiosyncratic, non-formal terms. Furthermore, the noise and scarcity of the imbalance dataset make this effort more difficult.

Detecting hate speech in streaming data is a difficult task. The truth is that most social media comments contain abbreviations and idiosyncratic, non-formal phrases, as well as weird characters. Furthermore, the volume of information, especially hate speech, grows rapidly by the second. This necessitates real-time processing and analysis of systems.

The contribution of the paper is three-fold. First, novel preprocessing strategies are proposed to increase the hate speech recognition model's accuracy. In these, using a Vietnamese abbreviation dictionary is a highlight approach. Second, PhoBERT is the proposed model with highly effective hate speech recognition. Finally, a hate speech detection system for online social media is constructed by using spark and kafka.

The ViHSD dataset has been chosen for experimentation and evaluation. The proposed hate speed detection system achieves an F1-score of 0.6888, which is higher than that of existing works on the ViHSD dataset. PhoBERT is used in a streaming system to collect and analyze social media data in real time.

2 Related works

In recent years, hate speech detection in Vietnamese has brought a lot of attention from many researchers. Quang Pham Huu et al. (2019)[2] proposed effective steps of text prepossessing combined with Logistic Regression to address the problem on the hate speech dataset provided by VLSP. Luu et al.(2021)[3] had a study on building the ViHSD dataset and hate speech detection in Vietnamese social media texts on this dataset. They experiment and evaluate their dataset on SOTA models. The highest performance is 0.6269 macro F1-score with m-bert cased.

Additionally, there have also been some research efforts for text classification tasks for Vietnamese social media texts. Khang Phuoc-Quy Nguyen (2020)[5] introduced different preprocessing techniques and key clause extraction with emotional context to improve the machine performance. Luan Thanh Nguyen et al.(2021)[6] proposed PhoBERT model as well as using traditional machine learning models and neural network models. PhoBERT model achieves more outstanding performance than the other models. Hence, PhoBERT model is proposed in our system based on the related work.

3 Proposed online social media hate speech detection system

3.1 Data Preprocessing

Based on the dataset attributes of comments on social media, the technical solution for text preprocessing stage work are proposed by following:

- Remove duplicate and NULL. Dataset includes 2553 duplicated comments and 2 NULL comments. Additionally, the dataset is divided into train, dev and test data. Thus, train, dev, and test data are combined and removed duplicate and NULL comments. Then using train_test_split to split train, dev and test data following rate 7:1:2 in the paper of ViHSD [3]. Table **??** shows number of labels in each class after removing duplicate and NULL comments.
- Lowercase all comments.
- Convert all characters such as ":)", ":3", ":(" to emojis.

- Remove all URL, mail, hashtag, mention tag.
- Remove all mixed words and numbers, such as "10k", "5tr", "3km".
- Remove all punctuation, special characters, and numbers.
- Remove all emojis.
- Normalize all repeated words such as haaaa, okkk, vlll.
- Normalize abbreviations. Creating a Vietnamese abbreviation dictionary from the dataset, which is the most essential aspect of the preprocessing step. Most abbreviations in each comment are manually replaced by origin words in the abbreviation dictionary. This dictionary contains 1725 terms chosen from train, dev, and test data. Moreover, the abbreviation dictionary also corrects a few of spelling mistakes such as liêm xỉ (correct: liêm sỉ), sa sỉ (correct: xa xỉ), sịn (correct: xịn). Mistaken words are not included in Vietnamese dictionaries or hardly appear in comments on social media. Table 2a shows some examples of Vietnamese abbreviation dictionary.

(a) Some examples of Vietnamese abbreviation dictionary

Abbreviations	Normalization
thíc, thick, thít, thik, thjx	thích
j, ji, rì, zì	gì
cutoe, dth, dthw, kewt	dễ thương

(b) Some examples with the same form but different meanings

Abbreviations	Meaning
ah	à
	anh
bn	bao nhiêu
	bạn
bt	biết
	bình thường

Table 1: Vietnamese abbreviation dictionary

- Predict abbreviations with different meanings but same form. Ridge is part of the Linear Regression family to be used as a model to predict abbreviations. Input is a comment, an abbreviation, start position and end position of the abbreviation in the comment. Output is the meaning of the abbreviation in the comment. Table 2b shows some examples with the same abbreviations but different meanings.
- Tokenize all comments. The space in Vietnamese is different to the space in English or common languages, is only the sign for separating syllables, not words. Tokenizing of texts is important, and it directly affects the results of models. Thus, ViTokenizer from the Pyvi library [5] is selected to tokenize all comments.

3.2 Deep learning models

Deep learning models using the most in natural language processing tasks are implemented for this task. Word embedding which helps learn representation for text is adopted before training models.

[5] https://pypi.org/project/pyvi/

Word embedding is a learned representation for text in which words with the same meaning are represented similarly. This approach to representing words and documents may be considered one of the key breakthroughs of deep learning on challenging natural language processing problems. fastText and PhoW2V are responsible for this stage.

- **fastText** is a multilingual pre-trained word vector, including Vietnamese, released by Grave et al. [1].
- **PhoW2V** is a pre-trained Word2Vec word embedding for Vietnamese, which is published by Nguyen et al. [4].

Deep learning models

- **Gated Recurrent Units (GRU)** is an advancement of recurrent neural networks. Compared to Long Short-Term Memory (LSTM), GRU is simpler but faster to train, and its performance is approximately equal to LSTM in several NLP tasks.
- **Long Short-Term Memory (LSTM)** is an advanced type of Recurrent Neural Network which is able to store information (long-term or short-term). It is capable of solving the vanishing gradient problem encountered by Recurrent neural networks.
- **Bi-directional Recurrent Neural Networks (Bi-RNN)** is a combination of two RNNs - one RNN moves forward, beginning from the start of the data sequence, and the other, moves backward, beginning from the end of the data sequence. Due to this mechanism, Bi-RNN can get information from the past (forward) and future (backward) states at the same time.
- **Bi-directionalGRU (Bi-GRU)** have the same mechanism as Bi-RNN but replacing RNN cells by GRU cells with Bi-GRU.
- **Convolutional Neural Networks (Text-CNN)** is a class of deep learning methods which has been dominating in computer vision tasks. CNN is composed of multiple building blocks but mostly are convolution layers, pooling layers, and fully connected layers.

3.3 Distributed Big Data AI Pipelines using Orca

BigDL is a Big Data AI project open-sourced by Intel. BigDL uses the Orca library located in the Analytics Zoo project, making it easy for data scientists to develop distributed AI applications.

The Orca library in BigDL seamlessly scales out the end-to-end AI pipeline from local laptops/PC to distributed clusters, as illustrated in Figure. 1. Specifically including: from data loading, to preprocessing, to feature engineering, to model training and inference:

- The user first installs BigDL and all the Python libraries (such as TensorFlow or PyTorch) in the local development machine (using pip or conda).

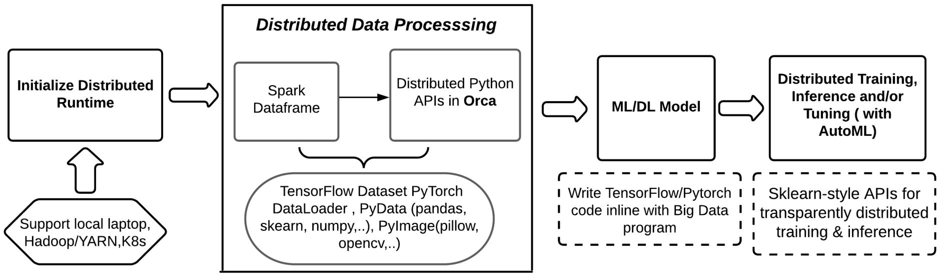
Distributed Data Processsing
Initialize Distributed Runtime
Support local laptop, Hadoop/YARN,K8s
Spark Dataframe
Distributed Python APIs in Orca
TensorFlow Dataset PyTorch DataLoader , PyData (pandas, skearn, numpy,..), PyImage(pillow, opencv,..)
ML/DL Model
Write TensorFlow/Pytorch code inline with Big Data program
Distributed Training, Inference and/or Tuning (with AutoML)
Sklearn-style APIs for transparently distributed training & inference

Fig. 1: Distributed AI Pipelines using Orca in BigDL.

- Loading the processes the data using a standard big data library (such as Spark DataFrames) or deep learning library (such as TensorFlow Dataset or PyTorch DataLoader); the Orca library runs these data processing pipelines in a data-parallel and distributed fashion.
- After that, defined the deep learning models in section 3.3 using standard deep learning APIs (that is, directly writing TensorFlow or PyTorch code inline with Spark program to build the model)
- Finally, the user uses the Estimator APIs in the Orca library for transparent distributed training, inference or Tuning (with AutoML) directly on Spark DataFrames using Spark.

3.4 Real-time hate speech detection system

Streaming data is the continuous flow of data generated by various sources, for example, from applications, networking devices, and server log files, to website activity, social media data, banking transactions, etc. By using stream technology, data streams can be processed, stored, analyzed, and acted upon as they are generated in real-time.

With instances of social media hate speech continuing to rise, the need for real-time monitoring of posts to quickly take action on the offensive is more important than ever. Under that pressure, social media platforms need to create tools to be able to process the huge volume of data created quickly and efficiently, especially to prevent hate speech.

In this research, Apache Kafka and Apache Spark are implemented which provide scalable, trusted, and low latency streaming platforms.

The architecture of the online social media hate speech detection system in this paper is illustrated in Fig. 2. Data is collected from social networks such as comments and classified using a proposed system for hate speech detection. Data after being classified is put into topics using Kafka produce. In this system, Kafka is used as a messaging system to connect components. Kafka Producer is responsible for writing data to topics, which is stored in distributed storage called Brokers. Spark structured streaming executes streaming data from Kafka

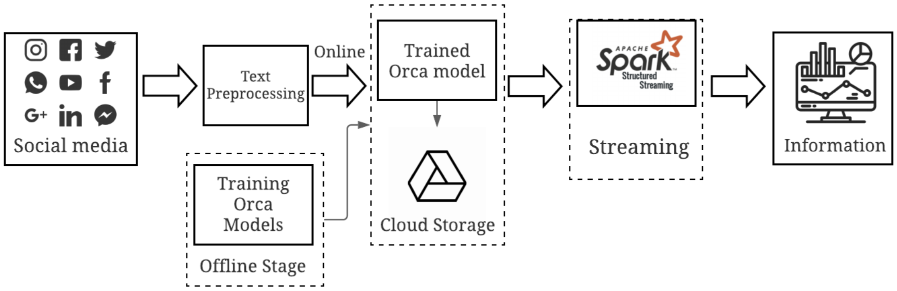

Fig. 2: The architecture of the online social media hate speech detection system.

topics to output sinks, which has a few formats like parquet, Kafka, console or memory, etc. Eventually, those classified comments are analyzed to gain insights, helping prevent hate speech.

4 Experiment

4.1 Dataset

No.	Comment	Label
1	Sao t gửi đc bây *(Why am I able to send it?)*	0
2	Ý thức còn ít hơn cả số tiền trong túi t *(Your level of civility is lower than the amount in my wallet)*	1
3	Quá ngu lồn đi =))) *(F*cking idiot =))))*	1
4	Im mẹ đi thằng mặt lon *(Shut the fuck up, you're a fucking c*nt)*	2
5	Bóng dơ *(Dirty queer)*	2

Table 3: The examples in ViHSD dataset.

In this paper, a few experiments are conducted on VLSP-HSD dataset [3]. This dataset contains 20345 comments collected from Facebook posts and comments. Each comment is labeled with one of three labels: CLEAN, OFFENSIVE, or HATE, which are labeled 0,1,2 correspondingly. The examples of dataset are describes in Table 3. The number of labels in each class is shown in Table 4. According to [3], the meaning of 3 labels are explained below:

CLEAN : There is no harassment in the comments.

Dataset	**0**	**1**	**2**
Training	12995	733	513
Development	3747	201	121
Test	1872	88	75

Table 4: Number of labels in each class

OFFENSIVE : The comment contains abuse and even profanity, yet it does not target any specific object.

HATE : The comments are harassing and abusive in nature, and are directed against an individual or group of people based on their traits, religion, and ethnicity.

ViHSD dataset has certain properties because it was crawled directly from user's comments on social networks:

- Having duplicate and NULL comments.
- There is a big skew between 3 classes.
- There are plenty of abbreviations or emojis, numbers and characters mixed in almost all comments.
- In comments, special characters such as &, % and # can be found.
- Some foreign language words can be found in the comments.
- Break marks such as a dot or a semicolon are not used consistently and clearly.
- Some comments contain unaccented letters.

4.2 Experimental results

(a) The experimental results of each model

Model	**F1-score**
LSTM + fastText	0.6643
LSTM + PhoW2V	0.6820
BiLSTM + fastText	0.6888
BiLSTM + PhoW2V	0.6748
GRU + fastText	0.6434
GRU + PhoW2V	0.6771
BiGRU + fastText	**0.6951**
BiGRU + PhoW2V	**0.6854**
CNN+fastText	0.6533
CNN+PhoW2V	0.6698

(b) The experimental parameters of each model

Model	**Parameters**
GRU+fastText	spatial drop=0.4, layer=1, gru units=160
GRU+PhoW2V	spatial drop=0.4, layer=1, gru units=160
LSTM + fastText	spatial drop=0.4, layer=1, lstm units=160
LSTM + PhoW2V	spatial drop=0.4, layer=1, lstm units=160
BiGRU+fastText	spatial drop=0.4, layer=1, gru units=160
BiGRU+PhoW2V	spatial drop=0.4, layer=1, gru units=160
CNN+fastText	128 Conv1D filter_sizes = 3, num_filters = 32
CNN+PhoW2V	128 Conv1D filter_sizes = 3, num_filters = 32

Table 5: The experimental results and parameters of each model

Table 5a shows the final experimental results for each model. The parameters of each model are shown in Table 5b. To assess the performance of all models, the macro F1-score is used as an evaluation metric. With an F1-score of 0.6951 (with fastText) and 0.6854 (with PhoW2V), BiGRU outperforms the other models. The results with word embedding are not significantly different for deep neural network models, shown in fig 3.

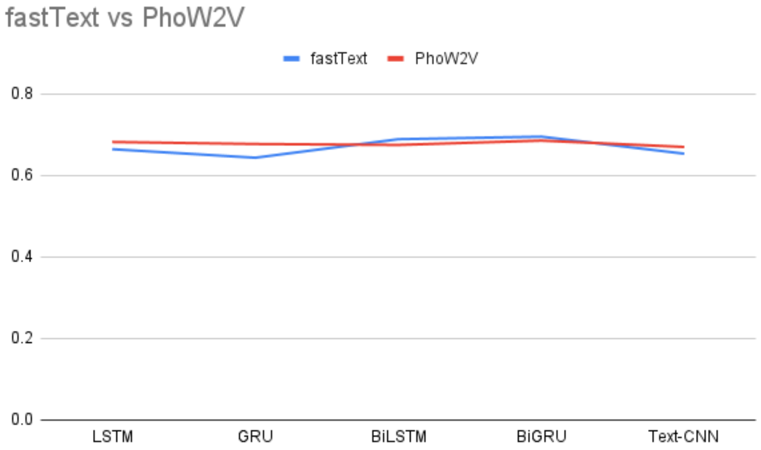

Fig. 3: Comparison of fastText's and PhoW2V's performance

4.3 Error analysis

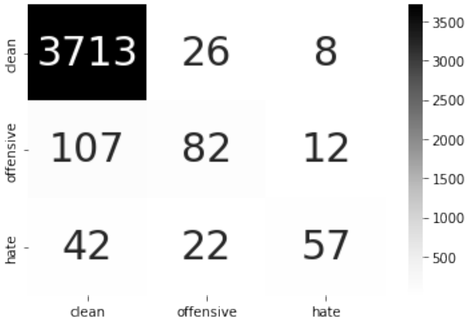

Fig. 4: Confusion matrix

Table 6: Some error cases

#	Comments	Predict	True
1	<person name> bị thế đéo nào được *(How the hell did this happen)*	1	0
2	ghê vãi *(how disgusting)*	0	1
3	như thằng điên *(mad guy)*	2	1
4	xạo lồn *(c*nt liar)*	1	2
5	già mất nết quá *(D*mn that spoiled old man)*	1	2
6	con gia nay chuyen gioi ho nhin ghê quá *(how disgusting, biddy is transgender)*	0	2

Table 6 presented some wrong prediction samples in ViHSD dataset. Firstly, the comments number 1 had "đéo", "ghê" having both negative and positive meanings in Vietnamese and our model can not understand the context of the comments. Additionally, the comment number 3 had "thằng" is a pronoun that causes confusion between labels 1 and 2. Following is the comment number 4 due to problematic data quality. The predicted result is correct, but the label is annotated incorrectly. Because the inter-annotator agreement for the dataset is just K=0.52 approximately threshold 0.5. Next the comment number 5, "già" here refers to old people in a compact and irregular form, not an adjective old this is a complexity of the Vietnamese social media texts. The comment number 6 almost does not have accented letters. Furthermore, as mentioned earlier, ViHSD is an imbalanced dataset, having as a result influenced significantly on the predicted results. Figure 4 shows the confusion matrix of PhoBERT. The correct prediction rate of label 0 is much higher than that of labels 1 and 2.

In summary, the deep learning models built on BigDL and the hate speech detection system for streaming data using spark and kafka on online social media are introduced in this study. With BiGRU, the best overall macro F1-score is achieved at 0.6951. Our work has some limitations, which will be discussed further below. Firstly, the dataset is unbalanced, which has an impact on predicted results. Secondly, due to language knowledge limitations, data is unable to adequately preprocess.

A set of stop words for hate speech dataset will be generated in the future, as well as solve the imbalance problem between classes in the dataset.

References

1. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. arXiv preprint arXiv:1802.06893 (2018)
2. Huu, Q.P., Trung, S.N., Pham, H.A.: Automated hate speech detection on vietnamese social networks. Tech. rep., EasyChair (2019)
3. Luu, S.T., Nguyen, K.V., Nguyen, N.L.T.: A large-scale dataset for hate speech detection on vietnamese social media texts. In: Fujita, H., Selamat, A., Lin, J.C.W., Ali, M. (eds.) Advances and Trends in Artificial Intelligence. Artificial Intelligence Practices. pp. 415–426. Springer International Publishing, Cham (2021)
4. Nguyen, A.T., Dao, M.H., Nguyen, D.Q.: A pilot study of text-to-sql semantic parsing for vietnamese. arXiv preprint arXiv:2010.01891 (2020)
5. Nguyen, K.P.Q., Van Nguyen, K.: Exploiting vietnamese social media characteristics for textual emotion recognition in vietnamese. In: 2020 International Conference on Asian Language Processing (IALP). pp. 276–281. IEEE (2020)
6. Nguyen, L.T., Van Nguyen, K., Nguyen, N.L.T.: Constructive and toxic speech detection for open-domain social media comments in vietnamese. Lecture Notes in Computer Science p. 572–583 (2021). https://doi.org/10.1007/978-3-030-79457-6$_4$9, http://dx.doi.org/10.1007/978-3-030-79457-6_49
7. Wiederhold, B.K.: Social media use during social distancing (2020)