An Implementation of Large Scale Hate Speech Detection System for Streaming Social Media Data

Long-An Doan*,†, Phuong-Thao Nguyen*,†, Thi-Oanh Phan*,†, Trong-Hop Do*,†

* University of Information Technology, Ho Chi Minh City, Vietnam.
† Vietnam National University, Ho Chi Minh City, Vietnam.

***Abstract*—The omnipresence of online social media brings various positive and negative consequences for society. Besides benefits, social media can cause big problem caused by hate and offensive contents. Detecting and removing those toxic contents using machine learning is a major research topic in social network. Two of the challenges of this topic are that the volume of social media data is so big and that these data need to be processed in real-time. In this paper, we set out to develop system to detect hate speech in Vietnamese YouTube comments using machine learning and big data technology. The streaming data from Youtube is processed in real-time using Kafka, Spark, and machine learning technology. Finally, a dashboard powered by Streamlit will be used to display the results.**

***Index Terms*—Streaming,Big Data,Hate Speech Detection, Social Network.**

I. Introduction

Social network is one of the technologies that entirely change the society. Besides many great benefits, social network also causes various problems of which online abuse is one of the most concerned ones. The most common type of online abuse is giving hate and offensive comments in social network. Even though users can report those comments, the ultimate solution is building a artificial intelligent based system to automatically detect and filter out these toxic contents from social networks. In recent years, hate speech detection is an active research topic that gained lots of attention from academy. However, current works on hate speech detection usually focuses on building machine learning models which yield better performance compared to that of previous ones. These works usually ignore the practicality of the system. For example, the models in current works are deployed in a local machine, which makes it impractical to scale up the system for processing huge number of comments from social network. Furthermore, for the hate speech detection results to be usable, they need to be obtained in real-time, which means streaming processing is mandatory for this kind of application. Very few, if any, current works concerns about the streaming processing for hate speech detection problem.

The practicality of the hate speech detection system is the main concern in this study. Therefore, instead of trying to modify machine learning models to improve the hate speech detection performance, this study focuses on presenting an implementation of a hate speech detection system for streaming social media data that is truly practical. That means that the system is capable of processing huge amount of streaming data from social network comments and producing results in real-time. To be more specific, comments from social network is collected and streamed through Kafka, which is a distributed streaming platform capable of dealing with huge amount of streaming data. The streaming data is then fed into a trained hate speech detection model integrated to Structured Streaming inside Spark, which is a powerful framework for big data processing. The use of Structure Streaming and Spark makes it possible to process a huge amount of social network comment and output the result in real-time. The result is then displayed through web-application based dashboards in real-time.

We shall provide related works in Section II and the system model in Section III of the remaining portion of the paper. Data and models are presented in Section IV. Then, in section V, we assess and analyze errors. Finally, in section VI, we draw a conclusion and suggest a course of action for future development.

II. Related Work

Under the growth of social media and the data explosion, hate speech detection is an urgent matter. There have been many studies on this issue across multiple languages, in which [1] proposed a method to identify Vietnamese data using the Bidirectional Long Short-Term Memory model, [2] combines the PhoBert pre-trained model and the Text-CNN model on the UIT-ViHSD dataset or [3] using traditional machine learning models such as SVM, Xboots combined with TF-IDF to identify on English data. In addition, the immediate updating and processing of data to be able to capture information or offer timely solutions such as weather data, and securities ... is also of interest. From there, applications and systems for this form of data were also

introduced. [4] processing real-time data from Twitter using Spark Streaming.

III. System model

The architecture of the proposed hate speech detection system for social media data is illustrated in Fig. 1. As can be seen in the figure, comments from social network is collected to create a source of streaming data. This data is streamed using Kafka and fed into components integrated inside Spark Streaming. These components includes preprocessing component, which is used to preprocess and clean the comment data. The clean comments are then fed into separately trained hate speech detection models. Since the model is integrated inside Spark Streaming, the hate speech detection results are produced in real-time. These results are then displayed on a dashboard created using Streamlit.

A. Frameworks for big data and streaming processing

We employ high-performance techniques that are frequently used in huge data processing applications in our detection model.

1) Apache Kafka: The distributed data streaming platform Apache Kafka has three key features. With its powerful and dependable storage, Kafka enables other systems to read and write data continuously. Kafka may also offer immediate data flow processing. Scalability, fault tolerance, and distributed support are all characteristics of Kafka. These capabilities allow Kafka to send enormous volumes of data in real time while maintaining data integrity. A server and a client make up the distributed Kafka system. Producers are programs that write data to Kafka, and consumners are programs that read and process that data. Producer and consumer are two distinct concepts in Kafka. In this report, Kafka-Python is used.

2) PySpark: Apache Spark is the powerful and well-known framework for big data processing. Spark can be deployed to run on a cluster of several machines that can be scale to adapt with the increase of input data. The advantage of Spark is not only that it can deal with big data but also that it can produce the output in real-time. Figure 2 illustrate the mechanism of streaming processing in Spark. Streaming data, which is the type of data that is generated continuously like comments from social network, are passed to models integrated inside Spark structure streaming and processed right after the moment it comes to produce results in real-time.

3) Streamlit: The tool was developed for Machine Learning Engineers, and it produces online interfaces similar to Jupyter notebooks. The fact that Streamlit does not need to display code is another unique feature of Jupyter notebook, enabling you to produce extremely polished products. In this post, Streamlit is used to build a dashboard that shows the outcomes. Figure 3 shows the system's output.

B. Model analysis of a system

We gather connections to videos with comments from the YouTube app. Use the *'bootstrap-server:9092'* gateway to send comments to Kafka's server after gathering them using the Google API. After collecting, the data will have a large number of data fields. The *'textdisplay'* data field - which holds the displayed text of the remark, will be filtered out.

Data is first stored in Kafka, after which it is integrated with Spark using the pyspark.streaming module and streamed to Spark.

We construct a predictive model pipeline based on the training set with more than 30,000 lines in order to be able to carry out the data preprocessing stages. Session IV of this procedure will go into more detail.

We utilize the load() method of the pyspark.ml module to store the trained model in spark. The label of the streamed-in comments will be predicted using this model.

Additionally, we employ the MapReduce technique for data visualization and statistics on the amount of labels for each type. The method of visualizing the results is improved by this approach.

IV. Dataset and model

A. Dataset

UIT – ViHSD dataset [5] introduced in 2021, a large-scale dataset for hate speech detection task on Vietnamese social media texts. The dataset contains 33.400 comments, collected from users' comments about entertainment, celebrities, social issues, and politics on different Vietnamese Facebook pages and Youtube videos. Each comment is assigned one of three labels: CLEAN (0), OFFENSIVE (1), và HATE (2). The inter-annotator agreement for the dataset is $\kappa = 0.52$ by Cohen Kappa index. We will use the dataset for model building and evaluation. Inside, we get 30.728 comments for the training and 2.672 for the testing. In addition, we extract 20% of the data from the train set to evaluate the model during the training.

B. Data preprocessing

With data collected from Facebook pages and YouTube videos, comments are diverse and complex that need to be processed to increase the classification ability of the model. We process data with commonly used techniques such as removing mentions, hashtags, URLs, emails, extra spaces in sentences, emojis, numbers, and characters. During this process, we consider removing the emojis because this may be the factor that makes the model biased. Then, we synchronize all characters in the text back to lowercase and filter the data by keeping the lines of data that the text contains.

On the other hand, the dataset is imbalanced between the data of the CLEAN label and the others that model

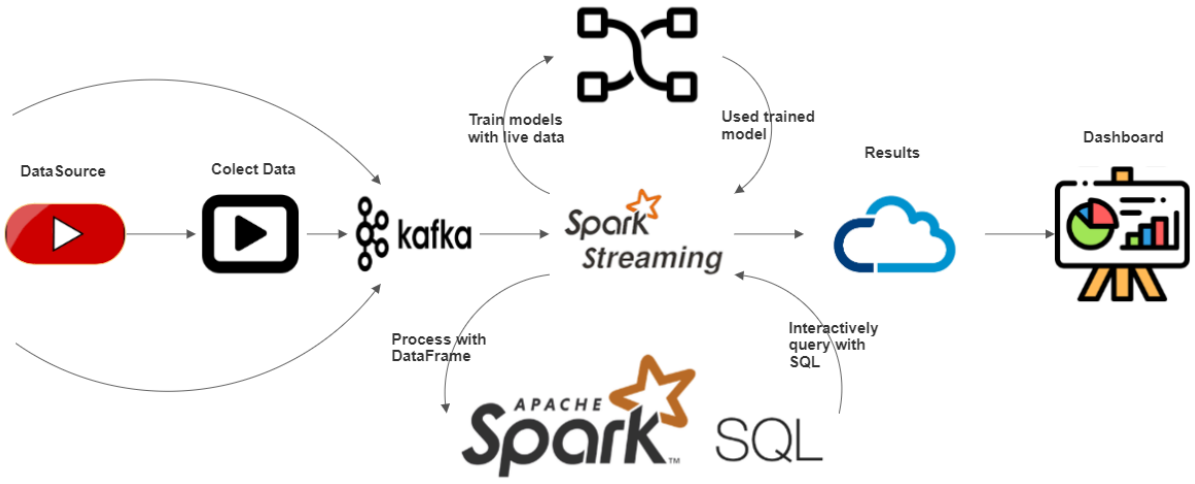

Fig. 1: System architecture

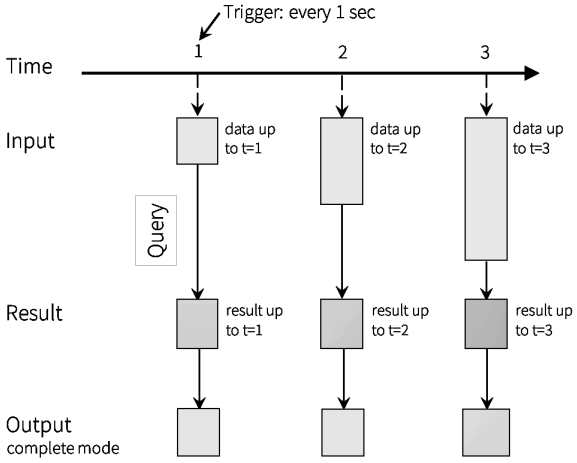

Fig. 2: Streaming processing mechanism in Spark

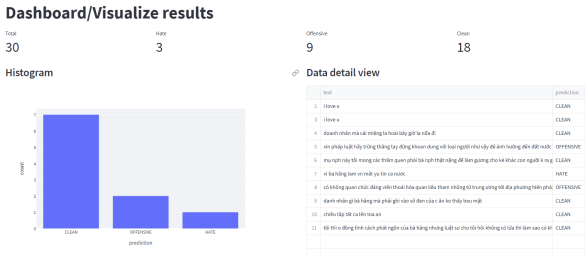

Fig. 3: Hate speech detection system's user-friendly dashboard

may be classified poorly on minority labels. So, we augment the dataset. So, we augment the dataset.

Next, we perform word separation and use two popular techniques in transferring text data to vectors: TF – IDF and CountVectorizer combined N-Gram.

1) Data augmentation: First, to handle the problem of unbalanced data with some labels in the ViHSD Vietnamese Hate Speech Detection dataset [5], we proceed with the data augmentation method presented by [6]. This data augmentation method uses four different methods to process:

1) Random Insertion (RI)
2) Random Swap (RS)
3) Random Deletion (RD)
4) Synonym replacement (SR)

Figure 4 describes the dataset before and after the data augmentation.

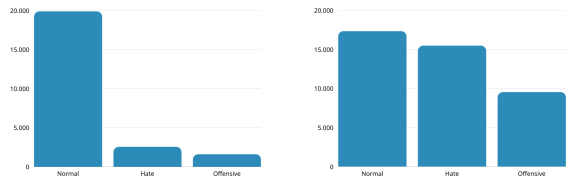

Fig. 4: Statistics of sentences by labels before and after data augmentation.

We conduct a boost on missing labels 1 (Offensive) and 2 (Hate). Then, we proceed to remove duplicate data points to obtain the dataset after strengthening. Besides, we have also conducted experiments on how effective it is to enhance the data of each label to get the best effect.

C. Model

1) Logistic Regression: Logistic regression is one of the most important and special algorithms in natural language processing that is the basic supervised machine learning algorithm for classification. Logistic regression estimates the probabilities of events, thereby predicting the probabilities of outcomes. We adjust the regParam parameters is 0.1 and 0.01.

2) Naive Bayes: Naive Bayes is a classification algorithm in a group of supervised learning methods. Based on Bayes theory, this method calculates the probability that the data points belong to a class. Then define the class of data points by choosing the class with the highest probability. This is an algorithm capable of training relatively quickly compared to other algorithms and is used a lot on practical tasks on a large dataset. In this paper, we set the smoothing parameter to là 0.0, 0.2, 0.4, 0.6, 0.8, 1 respectively.

3) DecisionTree: Decision tree is an algorithm that builds a classification model in the form of a structured hierarchical tree, used to layer objects based on a sequence of rules. After using the algorithm to train the model based on the train set consisting of its properties and classes, Decision tree will make rules to determine the class of data objects to predict. Compared to other algorithms, Decision Tree has few requirements in data preparation, data preprocessing as well as data normalization. With the model, we change the maxDept parameter with values such as 10, 20, 30, and default the other parameters.

V. Experiment and evaluate

The experimental procedure is illustrated in Figure 5

After preprocessing the data and building the model, we get the experimental results on Logistic Regression Naive Bayes Decision Tree models with two feature extraction techniques TF-IDF and CountVectorizer as the table below I.

Below, we provide the metrics used to evaluate the experimental results. The most common and widely used metrics for classification tasks in general, and for detecting hate and offensive comments in particular, are accuracy and average macro F1-score. Additionally , the average macro F1-score, which is the harmonic mean of Precision and Recall, is the most appropriate metric for this task due to the considerable unbalanced classes in the provided datasets (although processed). We have chosen to use the average macro F1-score (%) as the primary evaluation based on the results and the accuracy (%) as a supplementary evaluation.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad (1)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (2)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3)$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

TABLE I: The experimental results

		Accuracy	F1-score
TF - IDF	**Logistic Regression**	**0.8559**	**0.5911**
	Naïve Bayes	0.8270	0.5535
	Decision tree	0.8496	0.5112
Ngram+IDF	Logistic Regression	0.8532	0.5875
	Naive Bayes	0.7815	0.5623
	Decision tree	0.8496	0.5157

The highest results were obtained on the Logistic Regression model using TF-IDF with 85.59 (Accuracy) and 59.11 (F1-score). The naive Bayes model using TF-IDF correctly predicts many labels, but the classification ability of the model using CountVectorizer is higher. In the Decision Tree model, the two methods give almost similar results. The model's malicious language recognition is still low, but the results achieved are only 3.58% different on the F1-score measure and 1.09% on the accuracy measure compared to the baseline of the dataset [5]. This is also a fairly stable result because the models used are simple.

Based on this result, we analyzed the model with the highest results using the confusion matrix as shown in figure 6. We can see that the model has a high ability to correctly predict the class of the text. Here the model has high confusion between Offensive and Normal labels, Hate, and Normal labels. This is something we don't want in terms of using the model in our application such as identifying or removing malicious comments.

VI. Conclusion

In this paper, we present an implementation of a large-scale hate speech detection system for social network comments. The system was built using big data frameworks including Spark and Kafka. The contribution of the paper is presenting an implementation of the system that is more practical compared to previous works in this topic. Thanks to these big data frameworks, the system is capable of processing a large amount of comments from social networks and producing result in real-time. The streaming data of comments from Youtube is fed into the system and the results are displayed in a dashboard in real-time.

Acknowledgment

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

References

[1] H. T. Do, H. D. Huynh, K. V. Nguyen, N. L. Nguyen, and A. G. Nguyen, "Hate speech detection on vietnamese social media text using the bidirectional-lstm model," *CoRR*, vol. abs/1911.03648, 2019. [Online]. Available: http://arxiv.org/abs/1911.03648

[2] K. Q. Tran, A. T. Nguyen, P. G. Hoang, C. D. Luu, T.-H. Do, and K. Van Nguyen, "Vietnamese hate and offensive detection using phobert-cnn and social media streaming data," 2022. [Online]. Available: https://arxiv.org/abs/2206.00524

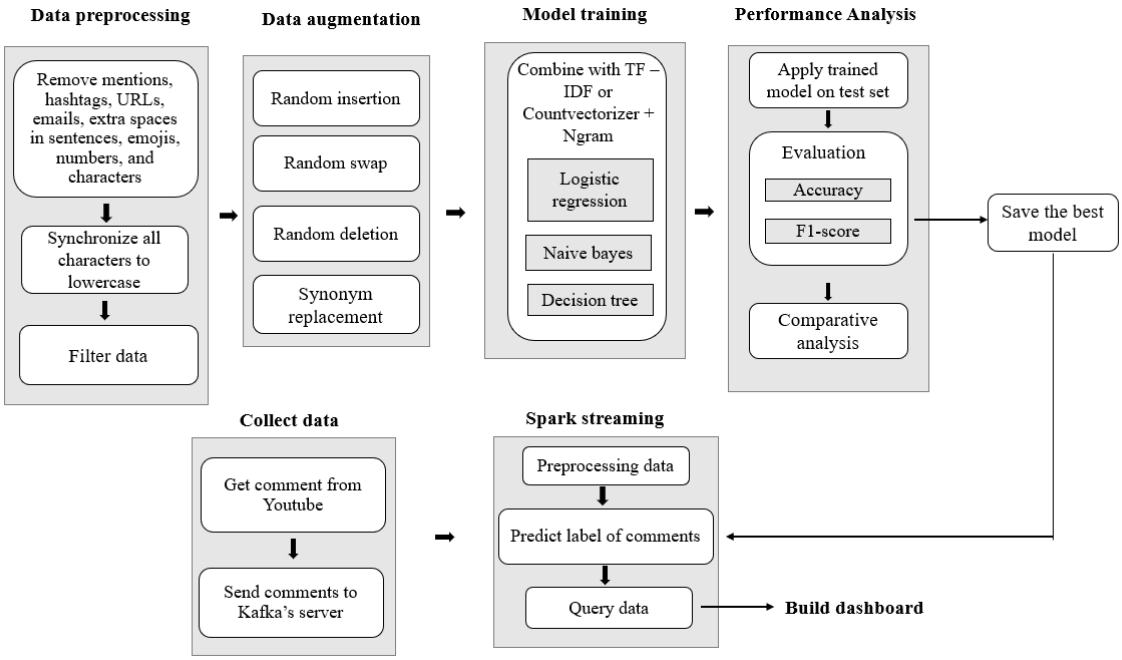

Fig. 5: Experimental procedure

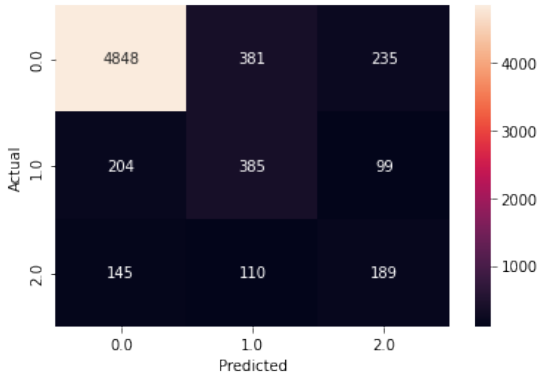

Fig. 6: Confusion matrix of Logistic Regresion model that uses TF-IDF

[3] A. Saroj, R. K. Mundotiya, and S. Pal, "Irlab@iitbhu at hasoc 2019: Traditional machine learning for hate speech and offensive content identification," in *FIRE*, 2019.

[4] N. Zaki, N. Hashim, Y. Mohialden, M. Mohammed, T. Sutikno, and A. Ali, "A real-time big data sentiment analysis for iraqi tweets using spark streaming," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1411–1419, 2020. [Online]. Available: https://www.beei.org/index.php/EEI/article/view/1897

[5] S. T. Luu, K. V. Nguyen, and N. L.-T. Nguyen, "A large-scale dataset for hate speech detection on vietnamese social media texts," in *Advances and Trends in Artificial Intelligence. Artificial Intelligence Practices*, H. Fujita, A. Selamat, J. C.-W. Lin, and M. Ali, Eds. Cham: Springer International Publishing, 2021, pp. 415–426.

[6] S. Luu, K. Nguyen, and N. Nguyen, "Empirical study of text augmentation on social media text in Vietnamese," in *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*. Hanoi, Vietnam: Association for Computational Linguistics, Oct. 2020, pp. 462–470. [Online]. Available: https://aclanthology.org/2020.paclic-1.53