Toxic Comment Detection on Social Media

Huynh Nhat Hao, Le Phan Thanh Dat, and Ha Nhu Chien

University of Information Technology

Abstract Toxic and offensive comments are ubiquitous problems on social media nowadays and result in plenty of influences on these platform users. Despite the fact that insulting language is not the majority in these media, it still harms several vulnerable types of users such as children. This leads to the demand of detecting toxic comments which are going to contribute to online platforms construction. We conduct our project on Vietnamese with the aim to enhance the online platform experience in case of minimizing Vietnamese offensive language. In this work, we have created a graphic user interface application that uses a trained model on a large-scale dataset for hate speech detection to automatically detect offensive comments on Facebook Pages and delete them, in order to create a better online environment.

Keywords: Toxic comment detection · PySpark · Social media.

1 Introduction

Social media comes to modern life with the hope that it is able to upgrade human connection through online communication. On the route to develop this kind of technological environment, toxic content is considered as an obstacle and it can increase dramatically along with the up-going number of posts, comments, and messages. A toxic comment is defined as a *rude, disrespectful, or unreasonable comment* that is likely to make other users leave the discussion, or conversely, it will trigger other users ego to join the endless discussion in a toxic way, and in the end, the users got nothing but the remorse feeling in their gut. Dangerously, toxic content is contagious like a disease in online environments.

In efforts to create clean social media environments, many works have tried to address the problem by creating large-scale datasets that facilitate the research community to concentrate on the modeling aspect. Some of the datasets in the Vietnamese language include the ViHSD dataset [1], ViCTSD dataset [2], etc. In this work, we make use of the ViHSD dataset to create a ready-to-use Graphic User Interface Application that automatically detects and deletes toxic comments on Facebook Pages[1]. Moreover, we built our application with the big-data picture in our mind, that is why we choose to use the Apache Spark [3] engine to process and modeling data, that way our application can easily scale up to process data parallelly on many workers on a cluster of computers.

[1] The project source code to build the application can be found here.

The content of the paper is structured as follows. Section 2 introduces and shows some analysis of the dataset we used to train our models. Section 3 presents our classification models using that dataset. Section 4 shows the results we obtained and compares them with the previous results. Section 5 describes our application design. Finally, section 6 concludes our works and proposes future works.

2 Dataset

The ViHSD dataset [1] contains 33,400 human-annotated comments about entertainment, celebrities, social issues, and politics from different Vietnamese Facebook Pages and YouTube videos. There are 3 labels in the original dataset: CLEAN, OFFENSIVE, and HATE. Table 1 shows some examples in three classes.

As defined in [1], A comment is considered clean, or safe if it has no harassment at all. The two classes offensive and hate are somewhat similar, they are both indicate comments that have harassment content. The only difference between offensive comment and hate comment is that offensive comment does not refer to any specific object while hate comment directly attacks an individual, an organization, or a specific object. Both two classes of comments are not safe when spread on the social network. We are only interested in detecting offensive comments, no matter it directly refers to any object or not, so we decided to merge two classes into one single offensive class.

The original dataset was severely imbalanced. Figure 1 shows the distribution of three classes in the train, dev, and test set.

Table 2 shows the experiment results from [1]. As expected, the F1-score is always much lower than the accuracy score for all models. The model that gives the best result is the one using bert-base-multilingual-cased pretrained model.

Table 1. Several examples of the ViHSD dataset.

#	Comments	**Label**
1	Thua !(English: Surrender!)	clean
2	Mất hình tượng vl(English: don't be a simp!)	offensive
3	KHON NAN(English: Damn)	offensive
4	Ông im mẹ mồm đi,xl quen mồm (English: Shut your f*cking mount off, your p*ssy liar.)	hate
5	Mấy thằng não cức(English: Your brain's full of sh*t)	hate

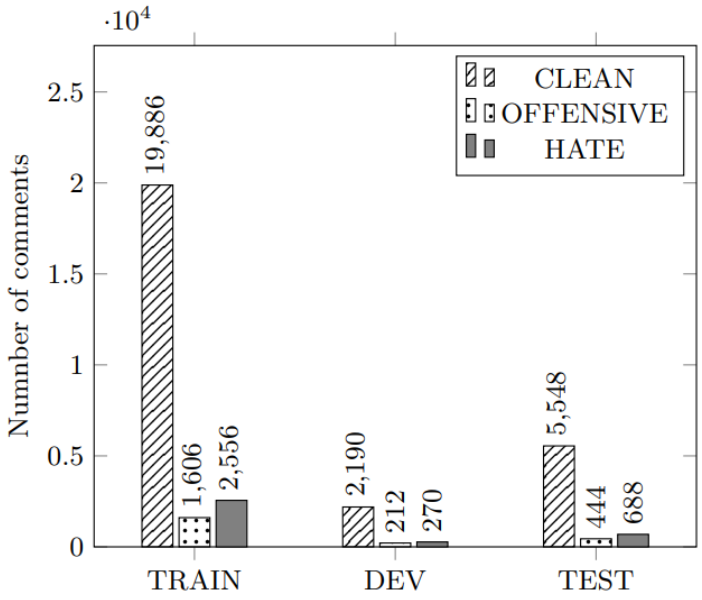

Figure 1. Classes distribution of the ViHSD datase, this figure was taken from [1].

Table 2. Experiment results of the ViHSD datase, this table was taken from [1].

	Models	**Pre-trained model**	**Accuracy(%)**	**F1-macro**
DNN models	Text CNN	fastText	86.69	61.11
	GRU	fastText	85.41	60.47
Transformer models	BERT	bert-base-multilingual-uncased	86.60	62.38
		bert-base-multilingual-cased	**86.88**	**62.69**
	XLM-R	xlm-roberta-based	86.12	61.28
	DistilBERT	distilbert-base-multilingual-cased	86.22	62.42

3 Method

First, we formally define the problem of text classification. Given a sentence with arbitrary length $x = [x_1, x_2, x_3, ..., x_n]$, where x_i are the tokens of the sentence, n is a positive integer, $n < max\_len$ where max_len is predefined. We wish to create a model that taking the sentence, produce an output indicate which class that sentence is belong to.

3.1 Logistic Regression

The first method we use is Logistic regression. As mentioned before, our problem only has two classes, the clean and offensive class. We use the TF-IDF features of the sentence to train the Logistic regression. We also use the oversampling trick to overcome the unbalanced dataset. The results obtained are in Table 3. Interestingly, when the oversampling trick is not applied, the F1-score of the minor class is 0, but we do not show that case in Table 3. This result reinforces the importance of pre-processing imbalanced datasets.

3.2 Bidirectional Long Short-Term Memory Network

The second method we use is a Bi-LSTM network, which is briefly described in Figure 2. In this method, we created our own vocabulary from the training dataset and use it to vectorize the input sentence. We used the SparkTorch Library[2] to create the Bi-LSTM model. SparkTorch is an API that help us integrate Pytorch model with Spark's ML Pipelines, we can load an existing trained Pytorch model and run inference on billions of records in parallel, thus SparkTorch help us get the best of both worlds.

4 Experiment Result

Our experiment results on the test set are shown in Table 3. As expected, the Logistic regression model using TF-IDF features and oversampling trick can not beat the Bi-LSTM model using oversampling trick on the F1-score. But what's interesting is, even the Bi-LSTM model without using oversampling trick chooses to classify everything it sees as class 0. When we apply the oversampling technique, the model has not even sacrificed any F1-scores on class 0 but can better predict class 1. Although it is not fair to compare our results to the result in Table 2 since their problem has three classes, ours has only two classes, however, our F1-score significantly outperforms the F1-score in the three classes case, and given that we are only interested in classifying two classes, that was a huge gain for us.

[2] More about the SparkTorch library can be found here.

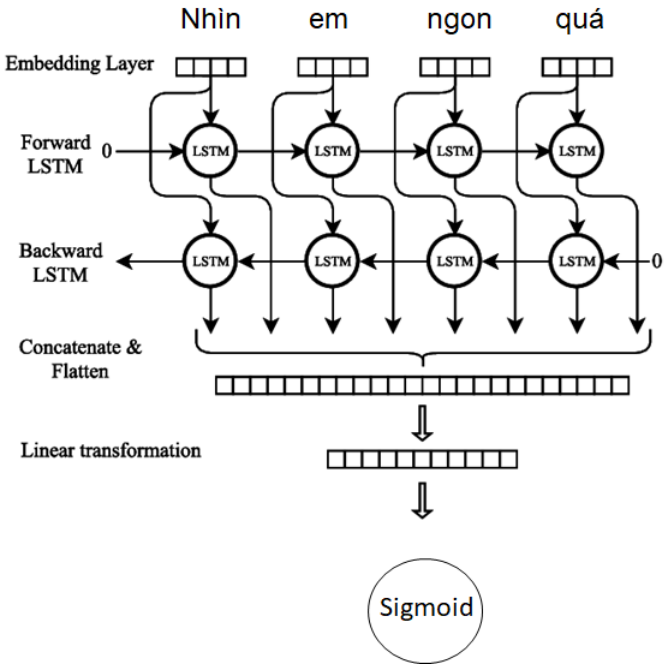

Figure 2. An Illustation of Bi-LSTM architecture. The input sentence after preprocessed will be tokenized into words, then vectorized using the vocabulary, then go through the embedding layer to become dense vectors. Next, two LSTM networks, a forward and a backward scan through the sequence of embedding vector and produce two last hidden states. Next, we concatenate the two last hidden states and apply a linear transformation on it, finally, we apply a Sigmoid function on it, produce the probability the model believed that the sentence is offensive.

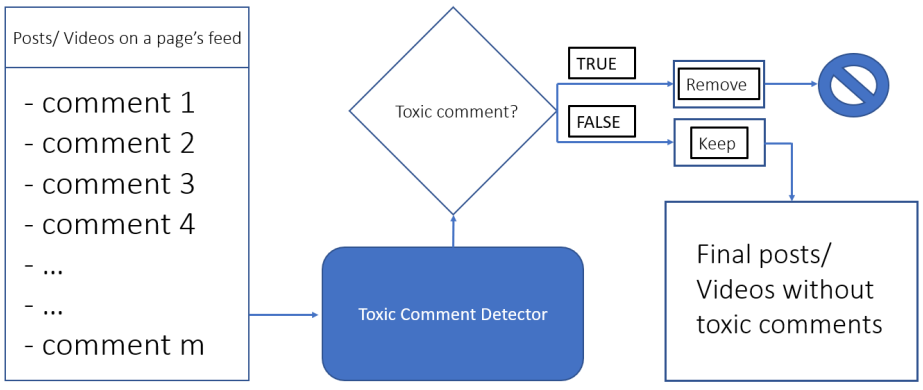

Figure 3. Workflow of the Toxic Comment Detection Application.

Table 3. Experiment results of the ViHSD dataset.

Models	F1 class 0	F1 class 1	F1-macro
SparkMLlib Logistics Regression + TF-IDF + OverSampling	0.88	0.49	0.68
SparkTorch Bi-LSTM	0.91	0.0	0.45
SparkTorch Bi-LSTM + OverSampling	**0.91**	**0.60**	**0.76**

5 Application Design

Taking one step further, we have made used of the above trained models to build a ready-to-use application that can detect and delete offensive comments on social media. We choose the Facebook Pages to be our application's target since Facebook provides us with the Graph API that facilitated us to pull all comments and delete them programmatically.

Figure 3 shows the workflow of our application. First, The user who owns a Facebook Page must log in to Facebook and grant us some essential permissions, which will allow us to interact with the user's Page content. Then, the users can pull all the comments on their Page, run inferencing with our trained models on those comments to detect offensive comments, and delete all those detected offensive comments with just three mouse clicks. Figure 4 shows a glance at our application interface.

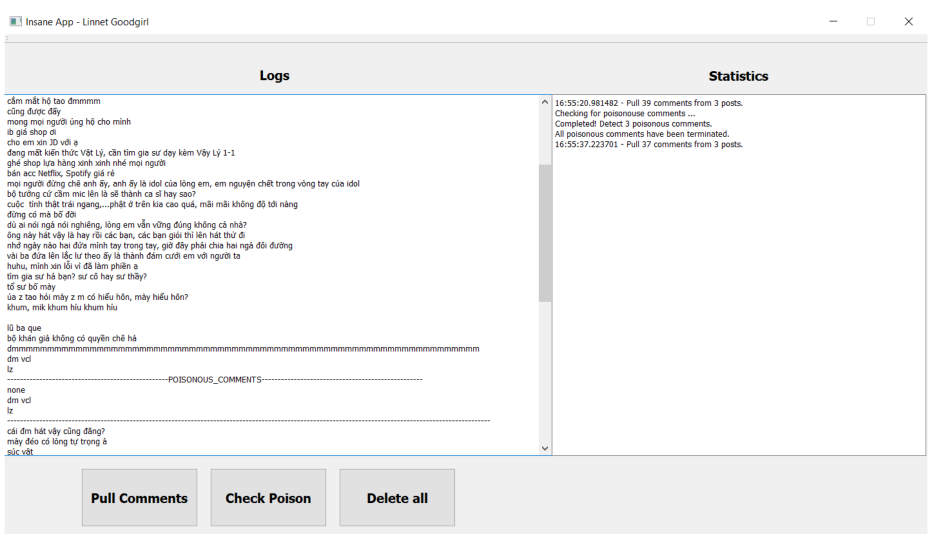

Figure 4. Our Application Interface.

6 Conclusion

In this work, we have created a Graphic User Interface Application that automatically pulls all comments from a Facebook Page, runs inferencing with our trained models to detect offensive comments, and deletes all those comments. Behind the scene, we trained our models using SparkTorch and Spark Mllib, which is the big-data platform that facilitates parallel processing and modeling. We used the Facebook Graph API to interact with the Facebook Page content. We acknowledge our limitation, include the F1-score of our model is still can be improved with more techniques. For future works, we will improve our model F1-score using more data, better model, and upgrade our application interface with more features and free to use for everyone.

References

1. Son T. Luu, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. A large-scale dataset for hate speech detection on vietnamese social media texts. *CoRR*, abs/2103.11528, 2021.
2. Luan Thanh Nguyen, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. Constructive and toxic speech detection for open-domain social media comments in vietnamese. *CoRR*, abs/2103.10069, 2021.
3. Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. Apache spark: A unified engine for big data processing. *Commun. ACM*, 59(11):56–65, October 2016.