A Distributed Architecture for Toxic Chat Detection

Wonhee Jung (wonheej2@illinois.edu) *University of Illinois Urbana-Champaign*, Kevin Mackie (kevindm2@illinois.edu) *University of Illinois Urbana-Champaign*, Cindy Tseng (cindyst2@illinois.edu) *University of Illinois Urbana-Champaign* and Conrad Harley (harley3@illinois.edu) *University of Illinois Urbana-Champaign*

***Abstract*—This paper presents the architecture, development, and use of a distributed system for toxic chat filtering. This system differentiates itself in that a) its filtering is based on machine learning and deeper contextual analysis, and b) it is deployed as a scalable and easily integrated web framework that can be adapted to any source of text for online interaction of any size. The platform is based on Docker and Kubernetes for easy deployment and dependency management, and uses state-of-the-art distributed systems technology to allow for fast scale-out to large systems. The system is presented in the context of a web chat application and Twitch chat bot as motivating examples.**

***Index Terms*—toxic comment, web, chat, detoxifier, detection, classifier, distributed, architecture**

I. Introduction

ONLINE platforms allow people to express their opinions freely, and stimulate collaboration across the globe. Unfortunately, online interaction may often come with loosened inhibitions in making profane, bigoted, or offensive remarks. We refer to such unwelcome remarks as “toxic chat”. Online systems may or may not have their own embedded profanity filtering, and those that do typically use pre-registered terms and simple pattern matching. This approach lacks the deeper contextual understanding needed to identify sentences that are toxic but that may not contain banned terms. Thus we propose a new toxic chat filtering system that differentiates itself in that a) its filtering is based on machine learning and deeper contextual analysis, and b) it is deployed as a scalable and easily integrated web framework that can be adapted to any source of text for online interaction of any size. The platform is based on Docker and Kubernetes for easy deployment and dependency management and to allow for fast scale-out to large systems. It uses state-of-the-art distributed systems technology for processing and storage, to allow for rapid scaling to any size while maintaining a shared file space (HDFS) between each Kubernetes Zone. This paper presents the architecture, development, and use of this system in the context of a web chat application and Twitch chatBot as motivating examples.

A. What we are going to make

We will create a prototype of PaaS/SaaS service that provides the following specific capabilities:

- machine-learning-based toxic chat identification and filtering engine
- integerated web chat application or chatbot that uses the engine to analyze a real-time stream of text

II. Toxic Comment Classifier

A. Training data

There is a dearth of labelled datasets for training classifiers to detect toxic comments. We conducted an online search and literature review on IEEE Xplore, Scopus, and Science Direct and concluded that the best dataset available is Toxic Comment Classification Challenge dataset released by Jigsaw and Google on Kaggle in 2018 https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge [1]

Example of a single citation [2]. Example of a two citations [3], [4]. Example of multiple citation [5]–[7].

B. Machine Learning Algorithm

TBD

III. Cloud system integration and development

The following technologies and solutions were integrated to provide a general framework that can scale to high volume/traffic in the future.

- Docker, Kubernetes - easy deploy and scale out
- CI/CD pipeline - to automate the build, integration, and deployment
- AWS, GCP, Heroku, etc - to deploy the solution into a mainstream PaaS infrastructure
- HDFS or similar - to store big data and share it between systems
- Scikit-learn or Apache Spark + MLlib - machine learing for the detox engine
- RESTful APIs - to help other applications integrate detox engine in the system

A. Deployment and scaling

- Docker, Kubernetes - easy deploy and scale out

B. Automated build, integration, deployment

- CI/CD pipeline - to automate the build, integration, and deployment

C. IaaS/PaaS infrastructure

- AWS, GCP, Heroku, etc - to deploy the solution into a mainstream PaaS infrastructure

D. Shared storage

- HDFS or similar - to store big data and share it between systems

E. Machine Learning Framework

- Scikit-learn or Apache Spark + MLlib - machine learing for the detox engine

F. Application Programming Interfaces

- RESTful APIs - to help other applications integrate detox engine in the system

Acknowledgment

The authors would like to thank . . .

References

[1] Jigsaw, "Toxic comment classification challenge," *https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data.*.

[2] M. Rybinski M., "On the design and tuning of machine learning models for language toxicity classification in online platforms," *Studies in Computational Intelligence*, vol. 798, pp. 329–343, 2018.

[3] L. Salminen J., "Neural network hate deletion: Developing a machine learning model to eliminate hate from online comments," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11193 LNCS, pp. 25–39, 2018.

[4] S. R.-G. Nestor Rodriguez, "Shielding google's language toxicity model against adversarial attacks," *http://arxiv.org/pdf/1801.01828v1*, 2018.

[5] R. K. Éloi Brassard-Gourdeau, "Impact of sentiment detection to recognize toxic and subversive online comments," *http://arxiv.org/pdf/1812.01704v1*, 2018.

[6] A. G. V. Spiros V. Georgakopoulos Sotiris K. Tasoulis, "Convolutional neural networks for toxic comment classification," *http://arxiv.org/pdf/1802.09957v1*, 2018.

[7] D. Noever, "Machine learning suites for online toxicity detection," *http://arxiv.org/pdf/1810.01869v1*, 2018.

Wonhee Jung (wonheej2@illinois.edu)

Kevin Mackie (kevindm2@illinois.edu)

Cindy Tseng (cindyst2@illinois.edu)

Conrad Harley (harley3@illinois.edu)