toxicity-filter

Github: https://github.com/minhd-vu/toxicity-filter

Authors: Vu Nguyen, Minh Vu

Introduction

In many online environments, toxicity can be a pressing issue regarding inclusivity, cyberbullying, and comfort. Toxicity can be defined as behavior that negatively impacts others and potential make the online environment less enjoyable. In order to remedy the issues that toxicity presents, a filter can be created to mitigate the impact of toxicity by detecting whether a statement is toxic or not and censoring it accordingly. Additionally, the filter should be able to discern content that is constructive criticism or from toxic content.

Input/Output

For example, we would like it so that when people are in a chatroom or a video game, if someone decides to type in a toxic statement, we would like to be able to detect that and immediately block it. However, we would not like to censor constructive criticism, so we must also consider differentiating between toxic speech and criticism.

Input	Envisioned Output	Actual Output
“You should kill yourself”	“you ***”	0.8508837223052979
“Play more defensively”	“play more defensively”	0.013167029246687889
“You’re trash!”	“you’re ***”	0.8076990246772766

Typically, words such as “kill” and “trash” are not censored by the filtering systems in place; however, because these tokens are deemed to make the statement toxic, they are censored.

Impact

Toxicity is most daunting in online gaming. Currently, most systems to deal with toxicity are simple curse word filtering, but the systems in place don’t go beyond to detect whether something is toxic or not. Moreover, many times some keywords may be censored when they should not be due to the lack of depth in the censoring software. By implementing a filter toxicity option, developers could utilize NLP in order to decrease the amount of toxicity.

Dependencies

```
mkdir data
cd data
pip3 install kaggle simpletransformers
kaggle competitions download -c jigsaw-toxic-comment-classification-
challenge
```

```
kaggle competitions download -c jigsaw-unintended-bias-in-toxicity-
classification
```

Training

```
python3 baseline.py
```

Running

```
python3 server.py
```

API

The API utilizes the regression model. Sending a POST request to the API will return a value of `0` to `1` depending on the level of toxicity that is detected. Create a `cache_dir` directory in the root directory or simpletransformers will throw an error. Now, `curl` a POST request to the API, just specify the message.

```
curl -X POST -H "Content-Type: application/json" \
    -d '{"message": "why are you so bad!"}' \
    http://localhost:5000/
```