CS410 Text Information Systems Final Project Documentation

Cong Chen, Jesse Deng December 2019

Overview of the function of the code:

For this project, we developed a tool that can help YouTubers filter out toxic comments under their video pages in a programmatic way.

The code we submitted includes two parts. First part is model training. We trained a classifier that can classify toxic comments and assign a label (either "0" – not toxic or "1" - toxic) to each comment. The second part is combine the trained model with YouTube API, so user (a YouTuber) can use the classifier to analyze comments under their video pages, pick those toxic comments and delete them automatically.

To use this tool, a user can directly open the Jupyter Notebook of the second part described above with the file name "toxic comments blocker for YouTube.jpynb" and run it. The detailed information of setup and how to run the software can be found below.

Software implementation:

Two labeled datasets were used for classifier training purpose. They are: 1) Davidson 2017 Twitter dataset, which includes > 24,000 tweets. The tweets have 3 labels, which are "hate", "offensive" and "neither". 2) Google Jigsaw training dataset, which includes > 150,000 annotated comments from Wikipedia Talk Pages. There are 6 labels for those comments. They are: "toxic", "severe toxic", "insult", "threat", "obscene" and "identity hate". Comments without any label are non-toxic comments.

We followed several steps before we can train a model.

Step 1: adjusted labels from two datasets to "1" – toxic or "0" – non-toxic. As we only need a "Yes" or "No" answer after analyze comments, we decided to train a binary classifier.

Step 2: balance the toxic and non-toxic comments in combined dataset. Before balancing, the ratio of toxic vs non-toxic comments are roughly 1:4, after balancing the ratio was adjusted to 1:1.

Step 3: feature generation. After preprocess the comment text data, we generated features for model training. The features include TF-IDF, POS tag, sentiment score and others.

Finally, we used the features prepared and trained two models. One is Logistic Regression, the other is Bayesian model. Both model performed similarly.

The second part of the software is combining trained with YouTube API and let user automatically classify the comments posted under their video pages and delete those toxic ones.

For this part, we wrote several functions to implement it.

- Request Comment. Get comments from YouTube Video page.

- Get next comment. Get comments from next page, as YouTube has 100 comments limit one time then save all comments in one list.
- Comment cleaning. Remove stop words, stemming and lemmatizing.
- Comment classify. Load the trained classifier and label comments.
- Comment delete. Delete those comments labeled as toxic.
- Main. Get YouTube credentials, link API and run functions defined above.

Usage of software:

The easiest way to use this software is to git clone the repository, open the "Toxic comment blocker for YouTube" Jupyter Notebook and run.

User needs to input his/her YouTube credential and the target video ID before running the code.

After running the code, the user needs to visit a URL provided to get an authorization code to authorize the software. Once the code in, the software will start to work, find and delete all toxic comments under the specified video page.

The change shows up after user refreshes the video page.

Please refer to our tutorial video for a demo of software usage. The video was uploaded through Coursera.

Contributions of team members:

Jesse Deng: team leader. Initiated the project idea and finished proposal. He finished second part of the software, which is combining the trained model with YouTube API to classify and delete toxic comments. He also prepared tutorial video.

Cong Chen: team member. Joined project idea discussion. He finished first part of the software, which is training classification models using public available datasets. He also prepared this documentation.