Documentation Milestone 2

April 27, 2019

1 Prediction of new data

The python script “predict.py” can be used to predict the labels (hateful or not) of new data with existing models.

Before you run it, you have to change the following input variables;

- data_path: Location of the new data.
- bert_model_path: Location of the BERT model.
- xgb_model_path: Location of the predictions model.
- bert_output_path = Location where the BERT features should be saved.
- predictions_output_path = Location where the predictions are saved.

The data_path should point to a csv file with a column called “Comment_text”.

BERT is a huge machine learning model and it takes quite long to generate features with it, that why the BERT features are stored. If for some reason an error happens afterwards, the BERT features don’t have to be computed again.

The output is a csv file with three columns: the original comment, the predicted probability that is is a hateful comment and the binary prediction (True if the predicted probability is greater than 0.5).

To execute the script you just need to change the variables in the “User Input” section and then run the script.

2 Training the Xgboost Model

The file train_optimize_xgboost performs a hyperparameter optimization, train the best model on the full train set and reports performance on the test set and then retrains the same model on all data (train + test).

It needs the following input:

- data_path: Location of a dataset with the BERT features. These features have to be created with the create_bert_dataset.py script. Because of the BERT models size it is a slow process to create these features and the xgboost model will probably retrained with higher frequency than the BERT model, so I separated these two tasks.
- number_of_runs: Number of models that are trained during the hyperparameter optimization.
- optimization_output_path: Location where the results of the hyperparameter optimization are saved.
- final_model_path: Location where the xgboost model is saved.

3 Creating the BERT features

The script “create_bert_dataset” is useful to create BERT features for a new model training set. As this is a slow process it is separated from the training of the xgboost model.

The script only needs three inputs:

- data_input_path: Location of the data. Expects a csv file with a “Comment_text” and a “Hateful_or_not” column.
- data_output_path: Location where the BERT features will be saved. This file is used as input to “train_optimize_xgboost.py”
- bert_model_path: Location of the BERT model.

4 Training the BERT Model

Part of the delivery is a BERT model that I have fine-tuned to your task already. You probably won’t need to retrain it ever. However, if you get a much bigger dataset in the future and want to train a new BERT model, you can do it with this script.

You have to provide two variables:

- data_path: Location of the data. Expects a csv file with a “Comment_text” and a “Hateful_or_not” column.
- model_name = Location where the model will be saved.

Finetuning a gigantic model like BERT takes a lot of time. It makes sense to use cloud computing resources to run this script. Make sure to exclude the test set from the BERT finetuning process, otherwise you’ll have data leakage.

5 Additional Files

There are two more python files “functions.py” and “bert_functions.py” which contain helper functions for the above scripts. You don’t have to touch them, they need to be in the same folder as the scripts though.