

Toxic Content Filtering



Tammy Metz

CSPB 4830:
Natural Language Processing

Problem Overview

Some people view the internet as their own personal playground where they are not actually interacting with other human beings. Because of this, many comments are toxic in some way (racist, sexist, foul language, etc.).

This project will classify comments to identify and filter toxic comments.

Motivation and Goals

Implement machine learning-backed identification of toxic comments to enable content moderation at scale.

Dataset Description

Source of the data: Jigsaw comment toxicity dataset.

Size and structure:

1. Training data containing 159,571 rows of labeled comments ('comment_text') with the following boolean features (using 0 or 1): ['toxic', 'severe_toxic', 'obscene', 'threat', 'insult', 'identity_hate']
2. Test data containing 153,164 rows of comments in one file and labels in a separate file, of which 63978 were correctly labeled and kept

Preprocessing or cleaning: Removed comments from the testing set if any toxicity label was -1

Approach Overview

Remove test comments that have any toxicity label of -1, remove stopwords, tokenize words, pad shorter comments

Use Tensorflow keras to create a model with a neural network containing an embedding layer and 2 dense hidden layers

Send cleaned test dataset to model and evaluate model performance on labeled test dataset

Clean data

Create GloVe embedding

Create model

Train model

Run model and Evaluate

Use pretrained GloVe embedding to create embedding layer

Send training dataset to model

```
embedding_layer = keras.layers.Embedding(input_dim=vocab_size, output_dim=embedding_dim, weights=[embedding_matrix], trainable=True)
model = keras.Sequential([
    embedding_layer,
    keras.layers.LSTM(256, return_sequences=False),
    keras.layers.Dense(50, activation='relu', kernel_regularizer=l2(0.001)), # Hidden layer with 200 nodes and ReLU activation
    keras.layers.Dropout(0.1), # prevent overfitting to training data
    keras.layers.Dense(200, activation='relu', kernel_regularizer=l2(0.001)), # Hidden layer with 200 nodes and ReLU activation
    keras.layers.Dropout(0.2), # prevent overfitting to training data
    keras.layers.Dense(50, activation='relu', kernel_regularizer=l1(0.001)), # Hidden layer with 200 nodes and ReLU activation
    keras.layers.Dropout(0.1), # prevent overfitting to training data
    keras.layers.Dense(1, activation='sigmoid') # Output layer for binary sentiment classification
])
```

Tools and Techniques

- NLTK English stopwords corpus
- GloVe (Global Vectors)
pretrained word embeddings
- Tensorflow with keras
- Bespoke LSTM model

Experiments and Analysis

- Shortening `max_sequence_length` by a factor of 10 dropped model training from around 20 minutes to around 5 minutes with no discernable drop in accuracy
- Experimented with various batch sizes from 8 to 256 and got best results with 128
- Experimented with between 3 and 8 epochs, and decided on 4
- Random shuffling the training data helped, but not as much as I hoped it would

Results

Splitting dataset...

Read in 153164 test comments

Kept 63978 test comments

Calculating overall toxicity scores...

Training and test sets tokenized and padded.

Model compiled.

Epoch 1/4

1122/1122  93s 83ms/step - accuracy: 0.9353 - loss: 0.4182 - val_accuracy: 0.9520 - val_loss: 0.1559

Epoch 2/4

1122/1122  87s 77ms/step - accuracy: 0.9560 - loss: 0.1466 - val_accuracy: 0.9534 - val_loss: 0.1517

Epoch 3/4

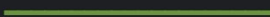
1122/1122  88s 79ms/step - accuracy: 0.9577 - loss: 0.1381 - val_accuracy: 0.9526 - val_loss: 0.1600

Epoch 4/4

1122/1122  92s 82ms/step - accuracy: 0.9574 - loss: 0.1313 - val_accuracy: 0.9521 - val_loss: 0.1516


Model trained.

Running trained model on test set.

2000/2000  36s 18ms/step - accuracy: 0.9254 - loss: 0.2259

Test Loss: 0.2223

Test Accuracy: 0.9267

2000/2000  37s 19ms/step

Results

Tag	Prec.	Recall	F-measure
nontoxic	0.9708	0.9473	0.9589
toxic	0.6016	0.7362	0.6621

Results

	nontoxic	toxic
nontoxic	54 691	3 044
toxic	1 647	4 596

(row = reference; col = test)

Challenges and Lessons Learned

- Keep booleans boolean!
- Don't need to keep a lot of words for comments (`max_sequence_length` can be short)
- I did not fully understand the API that came with this dataset and probably made things more complicated than necessary

Conclusions and Future Work

- Continue tweaking to improve precision, recall, and F-measure for toxic comments
- Experiment with pretrained models instead of training my own
- Try a transformer
- Try a larger GloVe embedding
- Try other embeddings