Kaggle Competition - Jigsaw Multilingual Toxic Comment Classification

Team Name: SEED=42

Jiayang Gao, Hongchen Luo

Task

- Cross-Lingual toxic comment classification
- Training Data on purely English data (2M)
- Validation on Foreign language data (8k, 3 languages)
- Test on Foreign language data (60k, 6 languages)

Difficulty

- Lack of understanding of foreign languages, hard to build vocabulary
- Lack of pre-trained cross lingual embeddings / models
- Lack of labeled foreign data (8k foreign data compared to 2M English data)
- Hard to define toxicity on foreign languages

Approach

- TF-IDF + Logistic Regression (0.88)
- Translation into English + Word Embedding + LSTM (0.9)
- Translation into English + BERT (0.91)
- Directly use Multi-Lingual BERT models (0.93)
- Use of validation set (0.94)

Details Our Framework

- Data Augmentation
- Model Structure
- Optimizer / Learning Rate
- Loss Function
- Ensemble
- Pseudo Labeling
- Stacking (ExtraTreeClassifier & StackNet)

Data Augmentation

- Translations (from original language to another language, and also translate it back to original language)
- Random sentence or word shuffle (swap)
- Random masking of original data (borrowed idea from BERT MLM training)
- Train Time / Test Time Augmentation

Special Data Treatment - 1

- We also came up with a somewhat special structure for tokenizing BERT data

[CLS] + [LANG] + [sentence tokens]

- [LANG] token is the abbreviation of the language of this sentence (en / fr / tr / ru, etc.), which can explicitly give the model information about which language this sentence is from.

Special Data Treatment - 2

- Putting different versions (translation/augmentation) of the same sentence in the same batch

- We found that it works better than random shuffle

- Another similar trick is sequence bucketing (which can speed up training), but we did not have time to implement

Model Structure

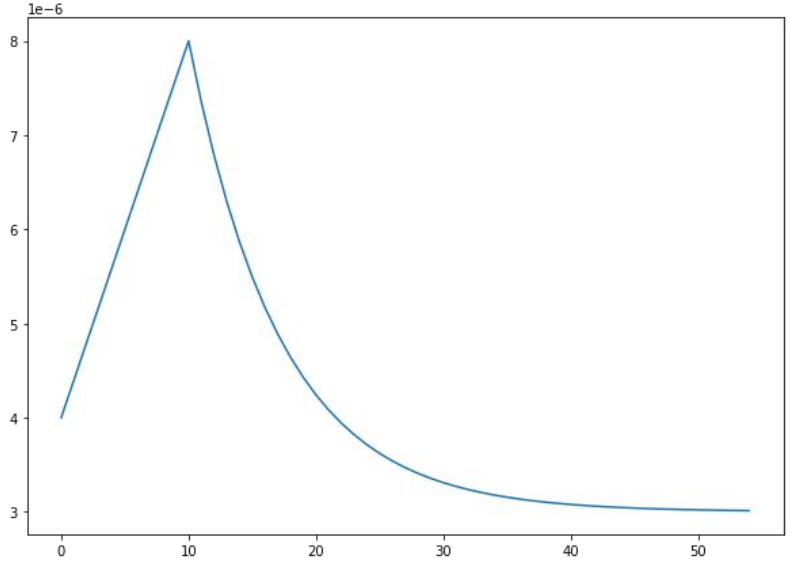
- Architecture of model (ResNet / RNN etc.)
- Dropout / regularization
- Mean/max pooling of hidden layers from BERT

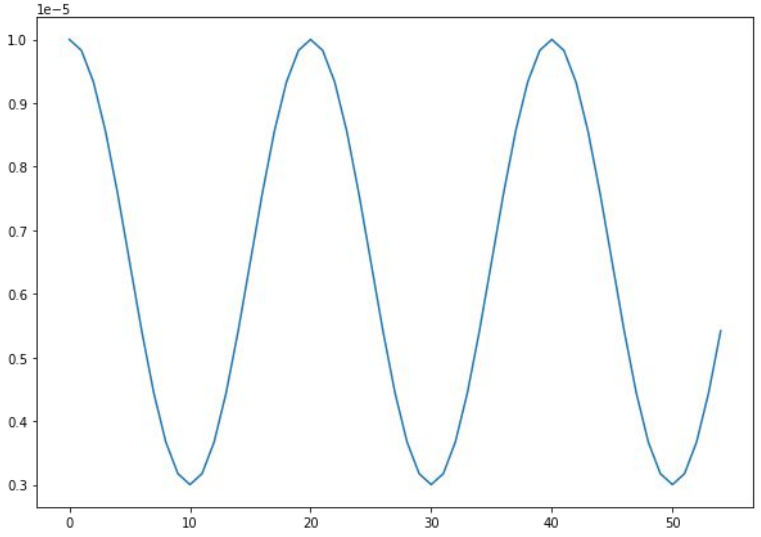

input_word_ids: InputLayer
input: [(?, 192)]
output: [(?, 192)]
tf_roberta_model: TFRobertaModel
input: (?, 192)
output: ((?, 192, 1024), (?, 1024))
global_average_pooling1d: GlobalAveragePooling1D
input: (?, 192, 1024)
output: (?, 1024)
global_max_pooling1d: GlobalMaxPooling1D
input: (?, 192, 1024)
output: (?, 1024)
input_laser: InputLayer
input: [(?, 1024)]
output: [(?, 1024)]
bert_and_laser: Concatenate
input: [(?, 1024), (?, 1024), (?, 1024)]
output: (?, 3072)
dropout_74: Dropout
input: (?, 3072)
output: (?, 3072)
new_dense_0: Dense
input: (?, 3072)
output: (?, 128)
dropout_75: Dropout
input: (?, 128)
output: (?, 128)
new_dense_1: Dense
input: (?, 128)
output: (?, 64)
deep_features: Concatenate
input: [(?, 3072), (?, 128), (?, 64)]
output: (?, 3264)
newout: Dense
input: (?, 3264)
output: (?, 1)

Optimizer / Learning Rate

- Adam
- Warmup and LR decay
- Cyclical / Cosine Annealing LR schedule
- Freeze Layers / Different LR per layer
- Early Stopping / Model Checkpoint

Optimizer / Learning Rate



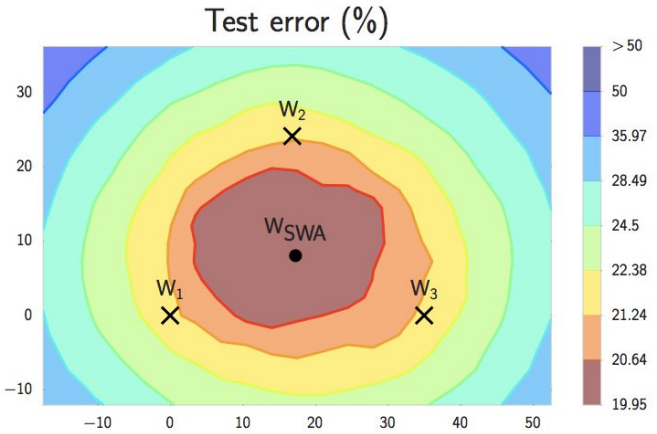


Loss Function

- Sample Weight (different per language)
- Class Weight (overweight positive)
- Binary Cross Entropy / Include Auxiliary Target

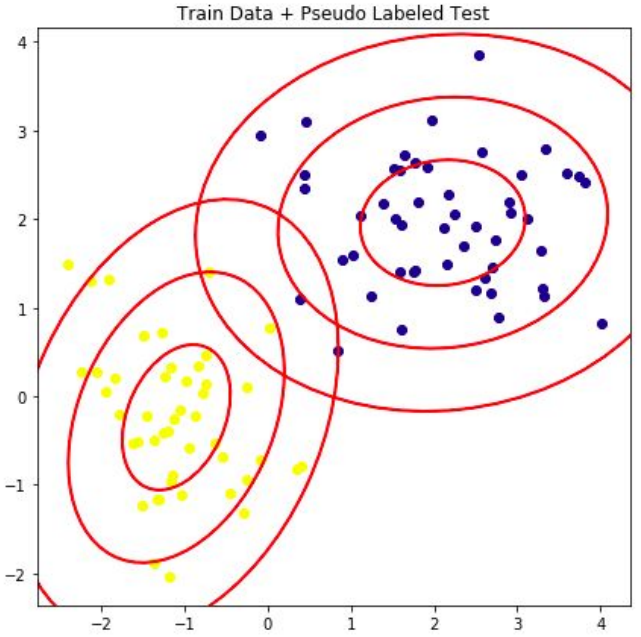
Ensemble

- SWA (Stochastic Weight Averaging)
- Snapshot Ensemble



Pseudo Labeling

- Use best prediction on the test set (6 foreign languages) as soft / hard labels



Stacking (with 1900 oof)

- CV: 5 fold with (toxic x language) stratified on 8k validation data

- Stacking Models (on top of base models OOF)
  - Logistic Regression
  - LGBM
  - ExtraTreeClassifier
  - StackNet (multiple layers of stacking)

- Main Difficulty: not enough data and unreliable CV. ExtraTreeClassifier helped build relatively reliable stacking.