Toxic Comment Detection and Classification

Stanford University CS229 Final Project Weiquan Mao Hao Li Hanyuan Liu

Introduction

- **Motivation:** Harassment and abuse are discouraging people from sharing their opinions. We aim to detect toxic comments in online conversations.
- **Problem Definition:** Develop machine learning models that can identify toxicity in online conversations.
- **Approach:** With Naive Bayes-SVM as our baseline model, we further implemented Bi-LSTMS, Bert models, and used two ensembling methods to improve quantitative results.

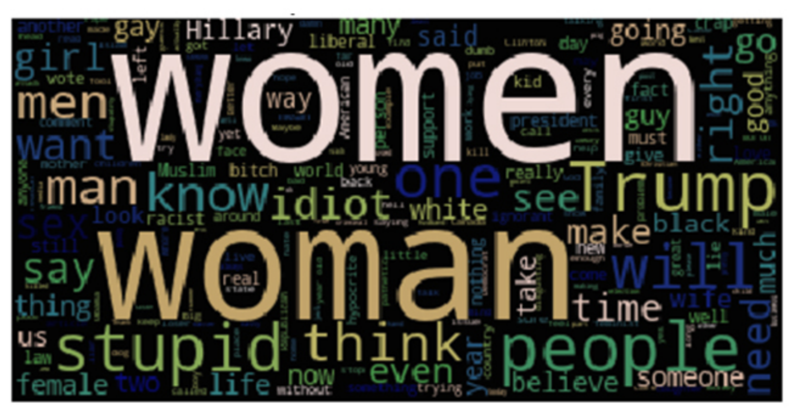
Data Sets

Civil Comments dataset:

The dataset comprises over 1804000 rows. Each row contains a general toxic target score from 0 to 1, a comment text, scores under various toxicity labels such as severe toxicity, obscene, identity attack, insult, threat, etc. The dataset is split into 80% as training set, 10% as dev set and 10% as test set.

Word Cloud Data Visualization:

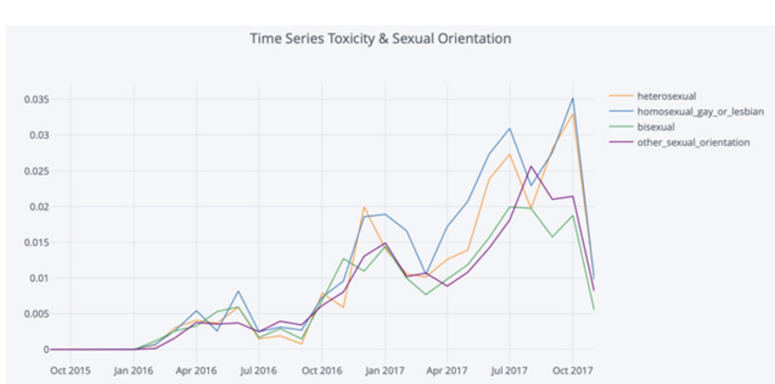
Top Words Frequency



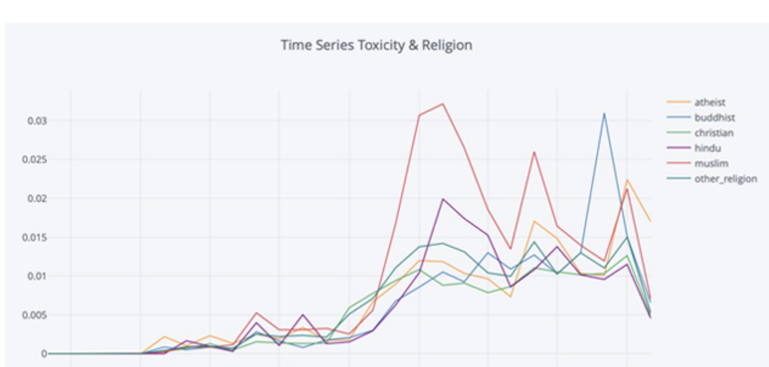
Female Related Words

Time Series Analysis:

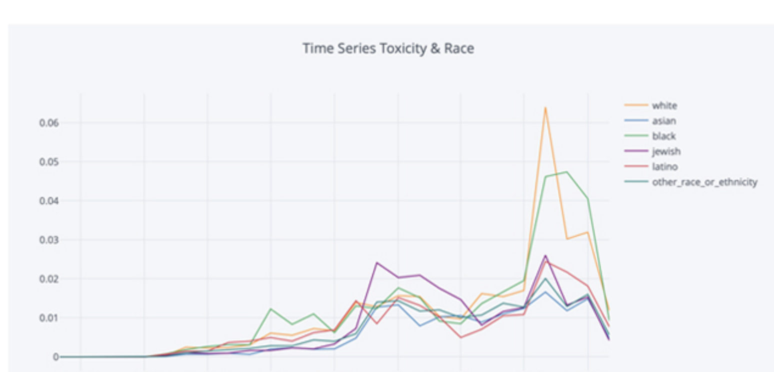
Toxicity: Disability



Toxicity: Sexual Orientation



Toxicity: Religion



Toxicity: Race

Approaches

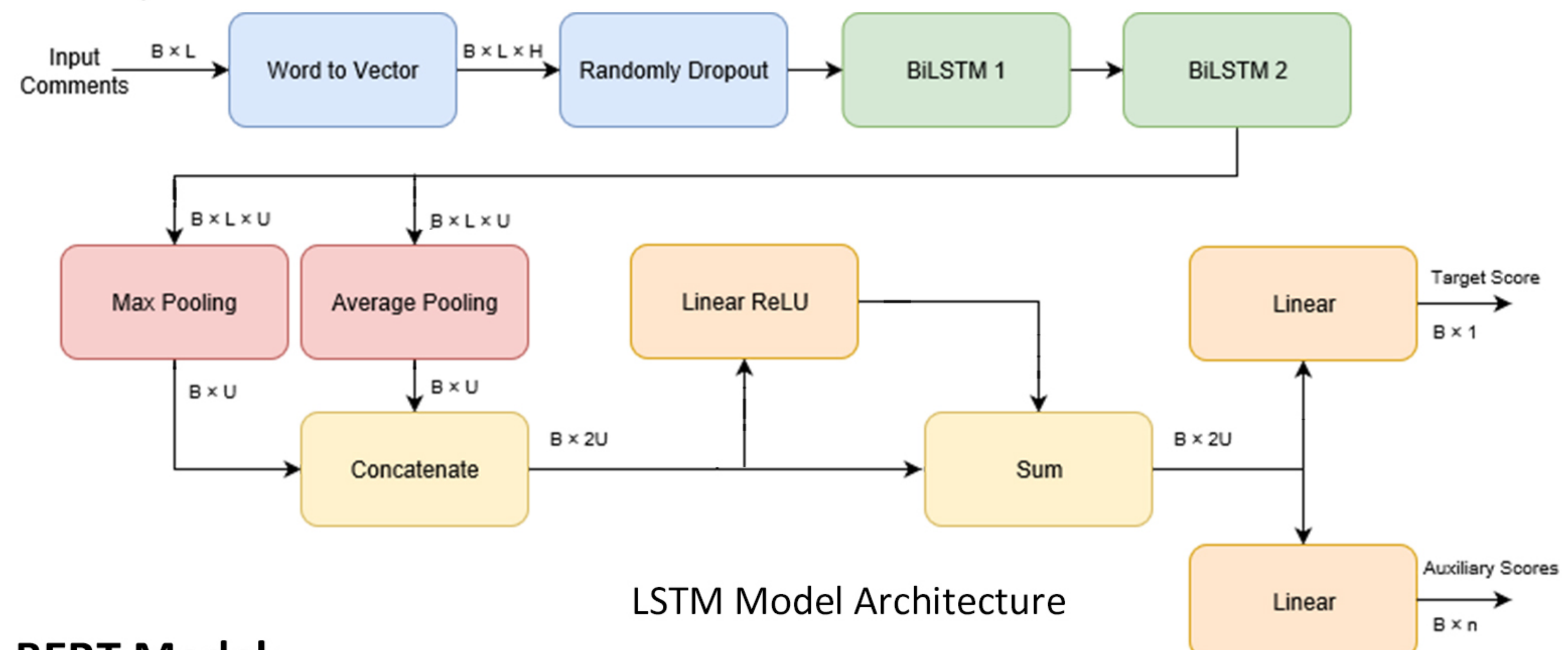
Baseline Model:

We combined Naive Bayes and Support Vector Machines to serve as our baseline model.

$$r_j = log(\frac{1+\sum_{i:y^i=1} f_j^i}{1+\sum_{i:y^i=-1} f_j^i}) \qquad y^k = sign(w^T x^k + b) \qquad x^k = r \circ f^k \qquad \min_{w,b} \frac{1}{2} w^T w$$

LSTM Model:

Seq2Seq architecture. We embedded the input text on the word-level. Then we added some drop-out layers to increase the robustness. 2-layer BiLSTMs with Max-pooling and Average-Pooling. At the end, in addition to the target score of toxicity, the model also predicted an auxiliary result.



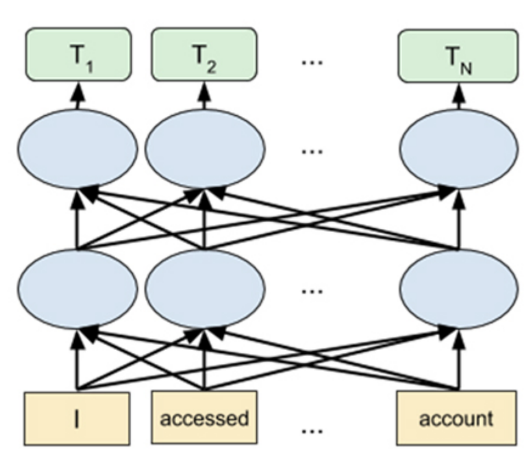
LSTM Model Architecture

BERT Model:

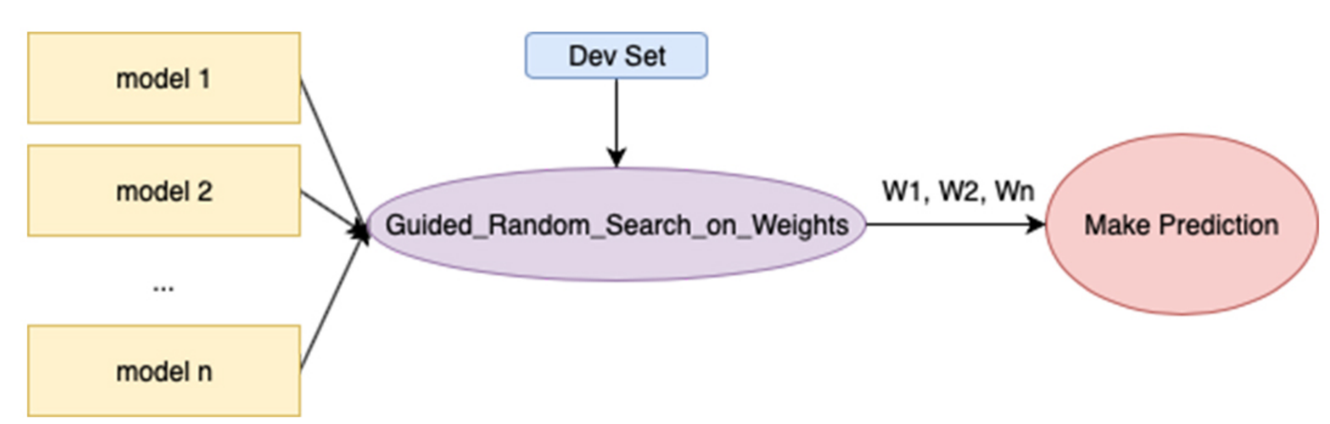
We used the pre-trained BERT-Base model, which is cased and has 12 layer with 768-hidden, 12-heads, and 110M total parameters. It can be fine-tuned with one additional output layer to create state-of-the-art models for sentence classification tasks.

Ensemble Method:

- Guided Random Search + Weighted Average Ensembling
- Follow the most confident prediction.



BERT(Google)



Random Search + Weighted Average Ensembling

Evaluation Method:

- Exact Match (the percentage of outputs that match exactly with the ground truth)
- F1 score (the harmonic mean of precision and recall)

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \qquad \text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \qquad \text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Results

Baseline Model:

The accuracy values of our baseline, Naive Bayes SVM, are shown below:

Description	Dev F1	Dev EM
Naive Bayes	68.33%	87.57%

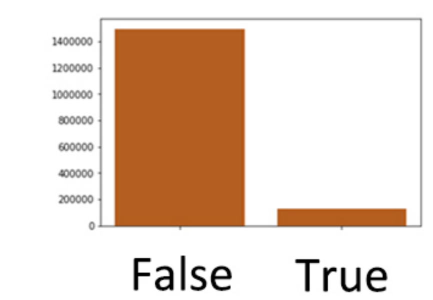
LSTM Model:

- We trained the LSTM model for 4 epochs using the Adam optimizer
- The initial learning rate is 1e-3 with a scheduler adjusting the learning rate.
- We used binary cross entropy loss as the loss function.

Weighted Loss LSTM Model:

To solve the data imbalance problem, we applied weighted loss to train our model: True-Positive ↑, False-Negative ↓

Description	Dev F1	Dev EM
Simple LSTM	77.95%	95.37%
LSTM with weighted loss pair(0.9, 0.1)	81.12%	95.21%



Contraction Mapping in LSTM Model:

Description	Dev F1	Dev EM
With contraction mapping	77.95%	95.37%
Without contraction mapping	76.04%	95.38%

BERT Model:

To train BERT mode, compare simple BERT with weighted loss BERT.

Description	Dev F1	Dev EM
Simple BERT	77.37%	95.73%
BERT with weighted loss pair(0.9,0.1)	81.19%	95.54%

Ensemble Method:

- Guided Random Search + Weighted Average Ensembling: (BERT with weight, LSTM with weight).
- Follow the most confident prediction: (BERT, BERT with weight, LSTM with weight).

Description	Dev F1	Dev EM
Best model without ensembleing	81.19%	95.54%
Ensembling with guided weight	81.57%	95.50%
Ensembling with most confident vote	84.28%	95.14%

Conclusions

- We tried three models on the toxicity classification problem.
- We used information from data visualization to preprocess data.
- Weighted Loss helped fix the problem of imbalanced data.
- Ensemble methods helped improve quantitative results.