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 ensemble 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.

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

\[ r_j = \log \left( \frac{1 + \sum_{x=1}^{2^n} f_{x_j}^i} {1 + \sum_{x=1}^{2^n} f_{x_j}^i} \right) \]

\[ y = \text{sign}(w^T x + b) \]

\[ x^* = \min \left\{ \frac{1}{2} w^T w, \frac{1}{2} n \right\} \]

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.

Approaches

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.

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

\[ F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \]

Results

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

<table>
<thead>
<tr>
<th>Description</th>
<th>Dev F1</th>
<th>Dev EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>68.33%</td>
<td>87.57%</td>
</tr>
</tbody>
</table>

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 \( \uparrow \), False-Negative \( \downarrow \).

Contraction Mapping in LSTM Model:

<table>
<thead>
<tr>
<th>Description</th>
<th>Dev F1</th>
<th>Dev EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple LSTM</td>
<td>77.95%</td>
<td>95.37%</td>
</tr>
<tr>
<td>Simple LSTM with weighted loss</td>
<td>81.12%</td>
<td>95.21%</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Description</th>
<th>Dev F1</th>
<th>Dev EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best model without ensembling</td>
<td>81.19%</td>
<td>95.34%</td>
</tr>
<tr>
<td>Ensembling with guided weight</td>
<td>81.57%</td>
<td>95.50%</td>
</tr>
<tr>
<td>Ensembling with most confident</td>
<td>84.28%</td>
<td>95.14%</td>
</tr>
</tbody>
</table>

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.