Diverse Ensembling with Bert and its variations for Question Answering on SQuAD 2.0

Stanford University

CS 224N Final Project | Winter 2019

Weiquan Mao Zhen Qin Zhining Zhu

Problem

The goal is to answer questions correctly given paragraph context from SQuAD 2.0. The target answer would be the span of text or N/A if there is no answer in paragraph. We use BiDAF as baseline, BERT-based architecture as the core, L1 regularization and other architecture changes on BERT. Ensembling method is also applied for improvement, which combines multiple models into a more robust Question Answering system by several different ensemble mechanisms.

Dataset/Task

We use SQuAD 2.0 as the reading comprehension data set. Every answerable SQuAD question has three answers provided. Dataset has been split into: 129941 training examples, 6078 dev examples, 5291 test examples.

Conclusions

After training 26 BiDAF-based, BERT-based models, and ensemble them with two algorithms, we push test F1 score to 78.841 and Test EM to 76.010. In conclusion, all our architectural changes increase the versatility of models and ensembling further amplifies such versatility in reducing training variances and achieving better performances.

Approach

1. L1 Regularization: \[ L(x, y) = \frac{1}{2n} \sum_{i=1}^{n} L(y_i, y^*) + \frac{1}{2} \|w\|_1 \]
2. Go "deeper": Add one more fully-connected layer to the output of BERT
3. Freeze shallow transformer layers: Freeze the weights of the first few layers of BERT (embedding layer, and the first few transformer layers)
4. Use BERT’s contextual embedding on BiDAF
5. Ensembling (a) Guided Random Search for Weighted Average Ensembling
   \[ p_{y_j} := p_{\text{start} \mid \text{pos}}(y_{\text{end} \mid \text{pos}}) \] predicted by the k-th model
   \[ \text{argmax}_{y_j} \sum_{y_i} w_k \times p_{y_j} \]
   (start_pos, end_pos) = \{w_1, w_2, ..., w_m\}
   (start_pos, end_pos) = \{max \text{start_pos}, max \text{end_pos}\}
   (start_pos, end_pos) = \{start_pos, end_pos\}

Experiments & Results

1. L1 Regularization: Train BERT with L1 regularization on weights of output classification and varies the coefficient. Increasing regularization strength helps improve the F1 and EM score.

<table>
<thead>
<tr>
<th>Regularization Coefficient</th>
<th>Dev F1</th>
<th>Dev EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Freeze</td>
<td>74.679</td>
<td>71.915</td>
</tr>
<tr>
<td>Embedding + 1 transformer layers</td>
<td>76.841</td>
<td>73.939</td>
</tr>
<tr>
<td>Embedding + 3 transformer layers</td>
<td>74.702</td>
<td>71.8</td>
</tr>
<tr>
<td>Embedding + 5 transformer layers</td>
<td>73.306</td>
<td>71.405</td>
</tr>
<tr>
<td>Embedding + 10 transformer layers</td>
<td>59.536</td>
<td>56.038</td>
</tr>
</tbody>
</table>

2. Freeze shallow transformer layers: Freeze BERT’s embedding layer and its first 1, 3, 5, and 10 self-attention transformer layers while fine-tuning. Freeze first 1 layer improves the original model.

3. Use BERT’s contextual embedding on BiDAF

The blue line represents training loss for BiDAF with BERT embedding. It converges faster than the original BiDAF model.

4. Final Ensembling Result
   - Ensembling is effective in decreasing variances and reducing over-fitting.
   - Quantitatively, our Ensembling model turns out to generalize well to the test set, with only a 1.3% decrease in both Test F1 and Test EM from Dev F1 and Dev EM, respectively.

Analysis

This plot visualizes the weights for the top 4 Ensembling models in one run(100 iters) of Guided Random Search of weights by plotting the distribution in histograms. Most of the models only bears a weight in the order of 0.001.

References


Guided Random Search for Weighted Average

Guided Random Search for Weighted Average

Follow the Most Confident Prediction

79.944 77.081 78.841 76.010

77.941 75.930 N/A N/A