Applying Ensembling Methods to BERT to Improve Model Performance
Charlie Xu, Solomon Barth, Zoe Solis
{cxu2, sbarth, zoesolis}@stanford.edu

Problem
Machine Comprehension (MC) is a complex task in NLP that aims to understand written language. Question Answering (QA) is one of the major tasks within MC, requiring a model to provide an answer, given a contextual text passage and question. It has a wide variety of applications, including search engines and voice assistants, making it a popular problem for NLP researchers.

According to the SQuAD 2.0 leaderboard, most high-performance models incorporate BERT in some way. All of the current top 18 submissions incorporate BERT in some way. However, there is much variation in the choice of ensembling and parameter tuning that can be done on top of BERT that differentiates much of the leaderboard.

Data/Task

Dataset: The Stanford Question Answering Dataset (SQuAD) is a large, diverse database of over 150,000 high-quality Wikipedia passages, reading comprehension questions, and accepted answers compiled by Stanford researchers. Roughly half of all questions are impossible to answer based on the given context. It uses the Exact Match (EM) and harmonic mean (F1) scores as metrics and maintains a leaderboard to see how the highest performing models compare against one another and against human performance.

Task: Use the provided context to produce an answer to the given question, or no answer if the question is impossible to answer. With the SQuAD dataset, all answers are selected to be subsets of the context, so the task can be reduced to finding the start and end indices of the predicted answer within the context.

We also used the SQuAD 1.1 dataset in the process of building our models. SQuAD 1.1 contains over 100,000 context paragraphs, questions, and answers, although it differs from SQuAD 2.0, in that all of its questions are possible to answer.

Context: “The descendants of Rollo’s Vikings and their Frankish allies would replace the Norse religion and Old Norse language with Catholicism (Christianity) and the Gallo-Roman language of the local people.”

Question: What was the Norman religion?

True Answer: Catholicism

Context: “Norman mercenaries were first encouraged to come to the South by the Lombards, to act against the Byzantines.”

Question: Who did the Normans encourage to come to the South?

True Answer: No Answer

Analysis

Comparison of Possible vs. Impossible Questions F1 Scores

<table>
<thead>
<tr>
<th>Model</th>
<th>Possible F1</th>
<th>Impossible F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>0.793</td>
<td>0.729</td>
</tr>
<tr>
<td>BeFC</td>
<td>0.804</td>
<td>0.739</td>
</tr>
<tr>
<td>BeT</td>
<td>0.877</td>
<td>0.864</td>
</tr>
<tr>
<td>BiT</td>
<td>0.877</td>
<td>0.861</td>
</tr>
<tr>
<td>Be+BeFC</td>
<td>0.850</td>
<td>0.812</td>
</tr>
<tr>
<td>Be+SSTQA</td>
<td>0.837</td>
<td>0.728</td>
</tr>
<tr>
<td>Be+BeFC+BeT</td>
<td>0.856</td>
<td>0.790</td>
</tr>
<tr>
<td>Be+BeFC+BeT-C-T</td>
<td>0.856</td>
<td>0.790</td>
</tr>
</tbody>
</table>

- Ensembling generally improved performance in both categories
- Addition of trimmed models improved ability to answer impossible questions, reduced ability to answer possible questions

Comparison of Question Type F1 Scores

<table>
<thead>
<tr>
<th>Question Type</th>
<th>Possible F1</th>
<th>Impossible F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>0.836</td>
<td>0.790</td>
</tr>
<tr>
<td>Specialized</td>
<td>0.804</td>
<td>0.762</td>
</tr>
</tbody>
</table>
- General models, except BiDAF, sometimes, improves performance
- Use of BeFC and BeTC-T/BeT-T tends to increase overall performance

Results

Overall Results

- Ensembling models, except BiDAF, sometimes, improves performance
- Use of BeFC and BeTC-T/BeT-T tends to increase overall performance

General vs. Specialized Model Performance

- General models, like Bert and BeFC, perform well on possible and impossible questions
- Specialized models, like BeTC and BeT, perform terribly on impossible questions, but extremely well on possible questions
- Use of classifier-based ensemble method should lead to better performance

Test Set Performance

- Given their high F1 and EM scores, we used our Be+BeFC+BeT-C-T model and Be+BeFC model on our test set, yielding these results.

Table 4: Model Test Set Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Test EM Score</th>
<th>Test F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Be+BeFC+BeT-C-T</td>
<td>73.556</td>
<td>76.721</td>
</tr>
<tr>
<td>Be+BeFC</td>
<td>73.221</td>
<td>76.454</td>
</tr>
</tbody>
</table>

Conclusions

- Ensembling boosts performance by leveraging the relative strengths of different models
- Manipulating training dataset led to significant differentiation of results between models, even those of same model architecture
- Use of a generally trained model as a classifier to determine when to use a specialized model can lead to significant increases in performance

Future Investigations

- Explore generating models specialized to predict impossible questions
- Train explicit NN classifier to classify questions as possible or impossible
- Implement other models and ensemble methods to make more ensemble models

Acknowledgements

Thank you to Sahil Aagrawal, Vivekumar Patel, and Dr. Chris Manning for their guidance and support!

References