Goals and Motivations

- Improve upon BDAF’s performance on a reading comprehension task with modifications such as using a GRU RNN, character-level embeddings, self-attention encoder, and using a QANet model.
- Create a model inspired by the Intra-Ensemble neural network (an end-to-end ensemble strategy with submodules which share some parameters) that uses our best performing model as a submodule.
- Explore this model’s effectiveness, and investigate how performance scales with number of parameters.

Motivations: Ensemble is effective but significantly increases number of parameters. How can we get the performance benefits of ensemble, with a less significant increase in parameter size?

Approach

- Intra: a language processing architecture that was largely inspired by the layer sharing technique proposed by the Intra-ensemble paper.
- Intra: has three submodules, with the first and second model sharing parameters for the first encoder block and the first and third model sharing parameters for the second encoder block.
- The submodules will use slightly different hyper-parameters and architectural components to increase the diversity of representations captured by each submodule.

Intra will use a traditional QA architecture as a submodule. We experimented with the following candidate submodules:

1. BDAF: Baseline BDAF model provided by CS224n class. Does not include character-level embeddings.
2. BGRU and Character Embeddings: Replaced the Bi-directional LSTM with GRU (Gated Recurrent Units).
3. Self-Attention Encoder: replaced the first encoder layer of the BDAF model with a self-attention layer that adds context-to-context attention and query-to-query attention.
4. QANet: new model that discards the recurrent concept of BDAF. Uses convolutions and multi-headed self-attention as encoder blocks.

We will deploy Intra on a reading comprehension task using the SQuAD 2.0 dataset (a collection of context, question, answer triplets).

After training the 4 candidate models we will:

- Evaluate each model’s EM, F1, AvAr scores
- Choose best models for Intra
- Diversity submodules through using different hyper-parameters, and architectures within the encoder blocks.


Experiments and Results

- Trained candidate submodules until loss, F1, and EM scores plateaued.
- Trained QANet with a learning rate of 0.1, hidden size of 100, drop probability of 0.1.
- All other submodules were trained with a learning rate of 0.5, hidden size of 100, drop probability of 0.2.

Our 4 submodel candidates achieved the following EM, F1, and AvAr scores:

<table>
<thead>
<tr>
<th>Model</th>
<th>NLL</th>
<th>F1</th>
<th>EM</th>
<th>AvAr</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDAF</td>
<td>9.11</td>
<td>60.08</td>
<td>54.53</td>
<td>67.22</td>
</tr>
<tr>
<td>GRU</td>
<td>5.12</td>
<td>57.22</td>
<td>72.67</td>
<td></td>
</tr>
<tr>
<td>GRU + Char Embeds</td>
<td>5.12</td>
<td>57.22</td>
<td>72.67</td>
<td></td>
</tr>
<tr>
<td>QANet</td>
<td>5.12</td>
<td>60.08</td>
<td>70.98</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Comparison of submodule performance on the test set.

- Model with GRU encoder performed best so chose it as submodule.
- For our Intra, the third submodule used an LSTM for its first encoder, increasing submodule diversity. Trained our Intra with different hidden state sizes.
- As a control, one of the submodules (GRU-50) was trained with a similar number of parameters as smallest Intra network.

Model | NLL | F1 | EM | AvAr |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BDAF</td>
<td>9.11</td>
<td>60.08</td>
<td>54.53</td>
<td>67.22</td>
</tr>
<tr>
<td>Intra-75</td>
<td>7.26</td>
<td>64.87</td>
<td>81.84</td>
<td>70.14</td>
</tr>
<tr>
<td>Intra-100</td>
<td>7.26</td>
<td>64.87</td>
<td>81.84</td>
<td>70.14</td>
</tr>
<tr>
<td>Intra-150</td>
<td>7.26</td>
<td>64.87</td>
<td>81.84</td>
<td>70.14</td>
</tr>
<tr>
<td>Intra-200</td>
<td>7.26</td>
<td>64.87</td>
<td>81.84</td>
<td>70.14</td>
</tr>
</tbody>
</table>

Table 2: Comparison of Intra models with different number of parameters (number to the left of Intra represents hidden size), and performance of a submodel with a comparable number of parameters.

The Intra model with a hidden size of 100 achieved the best dev set score, so we ran it on the test set. It achieved scores 65.46 (F1) and 62.40 (EM).

Analysis

- Using GRU RNN increase performance and decreased training time.
- Using character level embeddings improved the model’s handling of uncountered words.
- QANet had an unexpectedly low performance. Given the limited credit count and the extensive time needed to train the QANet (144 hours), we decided to focus on testing the Intra-ensemble model instead of debugging the QANet.
- Smallest Intra model allowed for better representations of the data than using a single model with a comparable number of parameters. This single model quickly overfit the data.
- Increasing the number of model parameters increased the performance of the model to an extent—increasing the hidden state from 75 to 100 increased the F1 and EM scores by 3.72 and 1.75, respectively.
- However increasing the hidden size from 100 to 150 hurt the models performance, lowering the F1 and EM scores by 0.56 and 0.64, respectively. We expect this was due to the model’s increased representational capacity not being utilized because there was an insufficent amount of data.

Furthermore, we split the Intra results by question type (How, What, Why, Which, Who, Where, When, OnHow), All three models performed the best on “when” questions and performed pretty on “how” and “why” questions. This observation makes sense intuitively, because “when” questions often have more straightforward answers than “how” and “why” questions if true.

Conclusion

- The following strategies performed better than a vanilla BDAF model: replacing the LSTM RNN with GRU RNN, including character embeddings, using self-attention instead of the LSTM RNN, and using QANet model instead of the BDAF model.
- Intra performed better than any of the models individually, even with models with approximately equal parameter sizes.
- Intra’s performance improved with more parameters to a certain-extent. After increasing the size too much, performance began to decrease.