Recent improvements in Q&A have seen a progression from using RNNs to CNNs due to improved training and inference speeds. The QANet model, introduced in Yu et al. (2018) [1], combines CNNs with self-attention, first seen in Vaswani et al. (2017) [2]. We build upon the BiDAF model described in Seo et al. (2016) to create our own implementation of the QANet model, achieving a single-model dev F1 score of 65.67, 4.48 points higher than the baseline BiDAF model [3]. We complement the QANet model with our own extension on the conditional output layer described in Kim and Wolff [4]. We achieve an ensemble dev F1 score of 67.08. Our ensemble model achieves a test F1 score of 63.33.

**Introduction**
- Early Q&A models relied on sequential end-to-end structure; however, more recent models propose more parallelizable structures.
- We create our own implementation of the QANet model. Our implementation achieves a similar performance score (61.03 F1) within an hour of training while it took the BiDAF baseline 2.5 hours to achieve 60.99 F1.
- We extend our implementation of QANet by implementing the conditional output layer described in Kim and Wolff [4] and then create our own conditional output layer.
- We further experiment with different novel changes on top of our baseline QANet model, including data augmentation, different model ensemble methods, and changing model sizes.

**Method**

| Baselines | BiDAF [3], BiDAF + character embeddings, QANet [1] |

| Dataset | SQuAD 2.0, F1, EM, Training Time |

**Improving QANet**

**Data Augmentation**
- Apply data augmentation by separately backtranslating context and answer from (context, question, answer) triple.
- Include backtranslated question/answer pair if new answer appears in new context.

**Cross-Conditional Output Layers**
- Based on Kim and Wolff [4], condition end probabilities on start probabilities and condition start probabilities on end probabilities (see Figure 1 for diagram of output layer).

**Ensembling**
- Implement segment and token max-ensemble, where possible answers are voted over their entire span vs. individual words.

**Results**

<table>
<thead>
<tr>
<th>Model</th>
<th>100%</th>
<th>80%-99%</th>
<th>60%-79%</th>
<th>Training Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiDAF</td>
<td>64.45</td>
<td>38.4</td>
<td>22.5</td>
<td>100</td>
</tr>
<tr>
<td>QANet</td>
<td>65.67</td>
<td>38.4</td>
<td>22.5</td>
<td>100</td>
</tr>
<tr>
<td>QANet [4]</td>
<td>63.53</td>
<td>38.4</td>
<td>22.5</td>
<td>100</td>
</tr>
<tr>
<td>QANet [4] w/ Data Augmentation</td>
<td>63.53</td>
<td>38.4</td>
<td>22.5</td>
<td>100</td>
</tr>
<tr>
<td>QANet [4] w/ Data Augmentation + bias</td>
<td>63.53</td>
<td>38.4</td>
<td>22.5</td>
<td>100</td>
</tr>
</tbody>
</table>

**Discussion**

**Data Augmentation**
- Improved performance for larger (i.e. x2 hidden size) models.
- Decreased performance for regular models.
- Poorly backtranslated answers introduce incorrect answer spans in the context which can result in poor performance.

**Forward-Backward Output Layer**
- Improved performance over our implementation of Kim and Wolff [4] by using conditional probabilities for start and end.
- Achieved lower overall performance than best model but improvement over [4] indicates their might be reason to continue exploring bi-directional conditionalities for the output layer.

**Ensembling**
- Four models (QANet, QANet+, QANet Avg., QANet Trunc).
- We saw overall improvement of 1.688 F1 from the baseline QANet model through segment max-ensembling.
- Both ensembling techniques leverage the individual strengths of each model, hence their improved performance.
- Segment max demonstrates improved performance over token max as it makes use of existing answers whereas token max could lead to it potentially creating an unseen response.

**Regularization and Layer norms**
- Attempted various regularization techniques, such as dropout, layer dropout, non-linear activations, L2 weight decay.
- Overfitting was still an issue, and occasionally became worse when some of these techniques were employed. Use of stochastic layer dropout meant that later layers (self-attention layer) would be dropped out more frequently than earlier layers (CNN layers), leading to loss of global interaction information and overfitting.

**References**


