SQuAD-ing Up as a Winning Team: Character Embeddings and Dynamic Coattention Networks

Kevin Guo (kyguo)°, Brandon Kang (bmkang)°
Department of Computer Science, Stanford University, Stanford, CA

Introduction/Background

- Question Answering is a task in the Natural Language Processing community that has received a lot of attention due to its broad range of implications for issues ranging from web search to data analytics.
- Question Answering has further increased in popularity due in part to the robustness and reasoning-based nature of the SQuAD dataset.²
- We first build upon the BIDAF model that contains only word-level embeddings by incorporating the Character-Level Embeddings as described in Seo et al. [2].
- We additionally implement Dynamic Coattention and create a model combining character-level embeddings with the Dynamic Coattention Network described in Xiong et al. [3].
- We measure the success of our implementations using EM and F1 scores and further experiment upon our models with hyperparameter tuning experiments.
- We find our model performs best with the Coattention and Character-Level Embeddings configuration with a learning rate of 0.6 and a dropout probability of 0.15 and associated EM and F1 scores of 59.189 and 62.787, respectively, on the test set.

Methods

1. Bi-directional Attention Flow (BiDAF) Model

2. Character-Level Embeddings
   - Pass preprocessed context/query words through a CNN to produce char embeddings.
   - Maxpool, concatenate to word embeddings, pass to Highway Network.

3. Dynamic Coattention Networks (DCN)

Results

<table>
<thead>
<tr>
<th>Model</th>
<th>lr</th>
<th>dropout</th>
<th>F1</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.5</td>
<td>0.2</td>
<td>60.74</td>
<td>57.39</td>
</tr>
<tr>
<td>Char-level Embs</td>
<td>0.6</td>
<td>0.2</td>
<td>60.86</td>
<td>58.11</td>
</tr>
<tr>
<td>Char-level Embs</td>
<td>0.5</td>
<td>0.25</td>
<td>61.92</td>
<td>58.47</td>
</tr>
<tr>
<td>Char-level Embs + Coattention</td>
<td>0.6</td>
<td>0.2</td>
<td>62.27</td>
<td>59.32</td>
</tr>
<tr>
<td>Char-level Embs + Coattention</td>
<td>0.6</td>
<td>0.2</td>
<td>61.78</td>
<td>59.06</td>
</tr>
<tr>
<td>Char-level Embs + Coattention</td>
<td>0.2</td>
<td>0.3</td>
<td>57.31</td>
<td>54.17</td>
</tr>
<tr>
<td>Char-level Embs + Coattention</td>
<td>0.6</td>
<td>0.15</td>
<td>63.91</td>
<td>60.43</td>
</tr>
</tbody>
</table>

Fig 1: Hyperparameter Tuning on Dev Set

Fig 2: Best Model (Coattention + Char-level Embeddings + Hyperparameter Search) Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>F1</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline on Dev</td>
<td>60.737</td>
<td>57.385</td>
</tr>
<tr>
<td>Char-level Embs + Coattention + Tuned Parameter - Dev</td>
<td>63.914</td>
<td>60.427</td>
</tr>
<tr>
<td>Char-level Embs + Coattention + Tuned Parameters - Test</td>
<td>62.704</td>
<td>59.138</td>
</tr>
</tbody>
</table>

Fig 3: Model Performance (EM and F1 scores)

Analysis

- Model pulled more data than necessary from the context; however, the extra information was also relevant to the ground truth answer.

Conclusions

- Adding character-level embeddings to the baseline BIDAF increased performance.
- Substituting the attention layer in BiDAF with coattention marginally increased performance.
- We were able to improve and achieve higher accuracy after performing a hyperparameter search.
- Our best model achieved an F1 score of 62.704 and an EM score of 59.138 on the test set.
- For future improvements, we are curious to see what would happen with a feedforward model that separates question types.
- We also suggest implementing data augmentation such as Easy Data Augmentation or more types of attention such as self-attention.

References