Methods

1. Character-level Embeddings
   Given \( (c_0, ..., c_L), \{q_0, ..., q_L \} \) words in the context and the question, respectively,
   \[ c_{char-context} = CNW(c_0, ..., c_L), \{q_0, ..., q_L \} \]
   \( c \sim \{c_{char-context}, \} \sim \{d_{word-context} \} \)

   a. Co-Attention Layer

   b. Self-matching Attention Layer
   The self-attention layer attention makes each word attend to all words in the context passage to get better knowledge of the context.
   \[ s_i^j = \tanh(W_d^d x_i^j + b_d) \]
   \[ a_i^j = \exp(s_i^j) / \sum_{j \in \text{S}} \exp(s_i^j) \]
   \[ c_i = \sum_{j \in \text{S}} a_i^j x_i^j \]
   i. Gated attention
   \[ a_i(W_0 q_i^c, c_i) + W_1 q_i^c \]
   ii. Sigmoid & linear transformation
   \[ a_i(W_0 q_i^c, c_i) \]
   c. Layer-Normed Scaled Dot Product Attention
   normalize the hidden state: \[ x_i^c = \text{LayerNorm}(x_i^c) \]
   obtain query, key, and value, and perform scaled dot product attention:
   \[ \text{Output} = \text{softmax} \left( \frac{QK^T}{\sqrt{d}} \right) V \]

Experimentation & Results

We explored how addition of character-level embeddings, attention mechanisms and tuning of hyperparameter would affect EM (exact match) and F1 scores using dev set.

<table>
<thead>
<tr>
<th>Method</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiDAF baseline</td>
<td>57.315</td>
<td>60.683</td>
</tr>
<tr>
<td>BiDAF + char embeddings + Co-attention</td>
<td>58.979</td>
<td>62.596</td>
</tr>
<tr>
<td>BiDAF + char embeddings + Gated Self-attention</td>
<td>58.327</td>
<td>61.472</td>
</tr>
<tr>
<td>BiDAF + char embeddings + Self-attention with transformation ( \alpha = 0.5 )</td>
<td>59.435</td>
<td>62.796</td>
</tr>
<tr>
<td>BiDAF + char embeddings + Self-attention with transformation ( \alpha = 0.25 )</td>
<td>61.822</td>
<td>65.157</td>
</tr>
<tr>
<td>BiDAF + char embeddings + Self-attention with transformation ( \alpha = 0.1 )</td>
<td>61.818</td>
<td>64.993</td>
</tr>
<tr>
<td>BiDAF + char embeddings + Self-attention with transformation hidden size ( \alpha = 150 )</td>
<td>58.931</td>
<td>62.107</td>
</tr>
</tbody>
</table>

We adopted BiDAF model with character-level embeddings and self-matching attention layer (with sigmoid & linear transformation) and learning rate = 0.25, hidden size = 100, batch size = 16 as our final model. The final model achieved performance of EM = 60.118, F1 = 63.866 on test set.

Analysis

1. We found that using sigmoid & linear transformation on attention layer would achieve better EM and F1 scores than gate mentioned in INKE paper [1].
2. Changing learning rate to 0.25 and 0.1 greatly improves performance compared to 0.5.

Conclusion

After training 8 BiDAF-based models with different designs of layers and hyperparameters, we push the dev EM to 61.822 and the dev F1 to 65.157, and subsequently push the test EM to 60.118 and the test F1 to 63.866. In conclusion, the best of our attempted architectural changes with fine-tuned hyperparameters results in better performance than the baseline.

Still, this research entails certain implications for future explorations, including the reasons of the transformation behaving better than the original gate and the rationale behind the unsatisfactory performance of co-attention and layer-normed scaled product self-attention, the latter of which we had to terminate during the training process due to its worse training pattern compared with the baseline.