Question Answering (QA) is an increasingly important NLP problem with the proliferation of chatbots and virtual assistants. In October 2018, Bi-directional Encoder Representations from Transformers (BERT) was released and achieved state-of-the-art results on a variety of NLP tasks, including QA. We seek to extend BERT with other performant QA architectures for SQuADv2.0.

**Motivation**

Question Answering (QA) is an increasingly important NLP problem with the proliferation of chatbots and virtual assistants. In October 2018, Bi-directional Encoder Representations from Transformers (BERT) was released and achieved state-of-the-art results on a variety of NLP tasks, including QA. We seek to extend BERT with other performant QA architectures for SQuADv2.0.

**Dataset**

Over 150,000 examples from 23,215 Wikipedia paragraphs in the following format: P: ... Bismarck was aware that public opinion had started to demand colonies for reasons of German prestige. He was influenced by Hamburg merchants and traders, his neighbors at Friedrichshuhr. The establishment of the German colonial empire proceeded smoothly, starting with German New Guinea in 1884. Q: Colonies were a sign of what amongst European countries? Answerable: TRUE A: prestige

Legend: P (paragraph); Q (question); A (answer)

**Solution Architecture**

**Bi-Directional Attention Flow (BiDAF)**

- Similarity matrix S (N×M)
- C2Q: weighted sum of question states → f
- Q2C: weighted sum of context states → g
- Output: stacked combination of f and g

**Dynamic Coattention Network (DCN)**

- Affinity matrix L ((N+1)x(M+1))
- C2Q: weighted sum of question states → a
- Q2C: weighted sum of context states → k
- 2nd level attention: weighted sum of k states → s
- Output: bi-LSTM encoding of stacked s and a

**Answer-Pointer**

- conditions and prediction on start prediction
- conducts two passes over modeling layer outputs with RNN/GRU
- second pass: uses final hidden state to facilitate end logits' dependence on start logits

**BiDAF Baseline**

- Answer-Pointer
  - weighted sum of context states
  - weighted sum of question states
- Similarity matrix S (N×M)
- C2Q: weighted sum of question states → f
- Q2C: weighted sum of context states → g
- Output: stacked combination of f and g

**BERT Integration**

To integrate the non fine-tuned BERT embeddings into our BiDAF and DCN implementations, we projected the BERT embeddings down, which were originally 768-vectors, to match the GloVe dimensions. Then, we pass in GloVe + GloVe ⊗ BERT, where ⊗ is the hadamard operator.

**Results**

<table>
<thead>
<tr>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-CASED</td>
<td>71.9</td>
<td>75.3</td>
</tr>
<tr>
<td>BERT-UNCASED</td>
<td>71.6</td>
<td>74.7</td>
</tr>
<tr>
<td>BERT-BiDAF</td>
<td>56.4</td>
<td>59.4</td>
</tr>
<tr>
<td>BERT-DCN</td>
<td>52.7</td>
<td>56.1</td>
</tr>
<tr>
<td>DCN</td>
<td>54.1</td>
<td>56.8</td>
</tr>
<tr>
<td>BERT-Answer-Pointer (RNN)</td>
<td>43.4</td>
<td>49.7</td>
</tr>
<tr>
<td>BERT-Lineare Answer-Pointer</td>
<td>67.7</td>
<td>71.1</td>
</tr>
<tr>
<td>BiDAF Baseline</td>
<td>55</td>
<td>58</td>
</tr>
</tbody>
</table>

**Error Analysis**

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>AvNa</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-CASED</td>
<td>74.0</td>
<td>53.8</td>
</tr>
<tr>
<td>BERT-UNCASED</td>
<td>66.0</td>
<td>38.5</td>
</tr>
<tr>
<td>BERT-BiDAF</td>
<td>68.0</td>
<td>38.5</td>
</tr>
<tr>
<td>BERT-DCN</td>
<td>52.0</td>
<td>38.5</td>
</tr>
<tr>
<td>DCN</td>
<td>48.0</td>
<td>7.7</td>
</tr>
<tr>
<td>BERT-Answer-Pointer (RNN)</td>
<td>54.0</td>
<td>61.5</td>
</tr>
<tr>
<td>BERT-Lineare Answer-Pointer</td>
<td>62.0</td>
<td>30.7</td>
</tr>
</tbody>
</table>

*selected 25 random examples to compare models

**Discoveries**

- **BERT-Linear-Answer-Pointer often attempts to answer unanswerable questions:**
  - This occurs due to magnified logits because of the lack of normalization when adding logits from linear and Answer Pointer layers
- **BERT-BiDAF/DCN vs BiDAF/DCN:**
  - BERT-X models converge faster to maximum EM and F1 than their non-BERT counterparts due to incorporating more contextual information via BERT’s FastText approach
- **BERT-Answer-Pointer suffers from early-summarization:**
  - arises due to RNN, which does not have the additional memory gate that LSTMs and GRUs have, hampering the ability to understand long-range dependencies

**Conclusions**

- **Non fine-tuned BERT embeddings can help speed up training in non-PCE implementations**
- **BERT-CASED is the most performant model; however, it’s main issue is attempting to answer unanswerable questions**
- **We need to develop better mechanisms to determine answerability**
- **True understanding of text is still a significant challenge**

**References:**