**Accelerated and Accurate Question Answering**

**Problem**
- The task of question answering on SQUAD 2.0 reading comprehension dataset has led many significant breakthrough models in building the machine comprehension system, important models like pre-trained contextual embeddings (PCE) model BERT and non-PCE model BiDAF.
- Limitation of PCE models like BERT:
  - need pretraining weights on a large-scale language modeling task
  - expensive to finetune due to the calculation resources we have.
- Non-PCE approaches are likely to underperform the PCE models by a large margin.
- Our project will focus on developing non-PCE model based on BiDAF and measure how well our machine comprehension system could understand the context.

**Dataset**
We will use SQUAD 2.0 dataset for this work, which comprises 129,941 training examples, 6078 dev examples and 5921 test examples, where each example is presented as context - question pairs. The target is whether the question is answerable from the given context, and if so, we need to predict the answer span from the given context.

**Analysis**

**1. Char-CNN:**
- utilize subword information and effective in modeling OOV (out-of-vocabulary) words.
- Combine with word level embedding from Glove.

**2. BiDAF with Self-Attention:**
- look at other positions in the input sequence
- find other words with strong relation of the current word

**3. Prediction Layer:**
- apply Bi-RNN and concatenate the output of RNN to attention output for end index calculation.

**4. SRU:**
- preserves the performance of RNN-like structure
- enable the computation to run in parallel to promote the training efficiency

**5. Regularization:**
- Embedding Dropout: perform dropout on the embedding matrix at a word level.
- Activation Regularization(AR): penalize activations that are significantly larger than 0.
- Temporal Activation Regularization(TAR): penalize the model from producing large changes in the hidden state from time t to time t+1.

\[
AR = \alpha L_2(\theta_i) \\
TAR = \beta L_2(\theta_i - \theta_{i+1})
\]

**6. Augmented No-answer Prediction:**
- output the probability of whether this question can be answered

**Approach**

**1. Most Improvements: Self Attention + SRU + Char CNN**
- SRU could accelerate our training process by 2x and improve EM by 1.8, F1 by 2.5.
- Char CNN concat word embedding has significant impact in EM and F1.

<table>
<thead>
<tr>
<th>Model</th>
<th>EM</th>
<th>F1</th>
<th>Time(Hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiDAF</td>
<td>57.64</td>
<td>60.80</td>
<td>12.98</td>
</tr>
<tr>
<td>BiDAF + Self Attention</td>
<td>59.20</td>
<td>62.45</td>
<td>13.63</td>
</tr>
<tr>
<td>BiDAF + Self Attention + SRU</td>
<td>60.04</td>
<td>63.66</td>
<td>6.06</td>
</tr>
<tr>
<td>BiDAF + Self Attention + SRU + Char CNN</td>
<td>65.20</td>
<td>65.48</td>
<td>12.80</td>
</tr>
</tbody>
</table>

**2. Other Parameters:**
- Several structures of the augmented no-answer inputs by trying some combinations of attention results, self attention results and start/end raw representation and logit

**Results**

**3. Fine-tuning:**
- We perform an extensive parameter tuning on tunable parameters
- Below graph shows our final training graph compared with initial training
- Blue - final parameters, Red - initial parameters

![Graph showing training results]

**Tables:**

### Table 1: Performance of Different Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Base + AR (alpha = 0.1) + TAR (beta = 0.05)</th>
<th>Base + NA Pred[Att, End rep]</th>
<th>Base + NA Pred[Att, Start rep, End rep, Max logits]</th>
<th>Base + NA Pred[Att, Start rep, End rep]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EM</strong></td>
<td>61.12</td>
<td>60.09</td>
<td>59.49</td>
<td>60.12</td>
</tr>
<tr>
<td><strong>F1</strong></td>
<td>64.40</td>
<td>63.70</td>
<td>62.24</td>
<td>62.80</td>
</tr>
</tbody>
</table>

**CS 224N Final Project | Winter 2019  Kuangcong Liu , Haoshen Hong , Xiao Wang  {cecilia4, haoshen, xiao1105}@stanford.edu  Computer Science Department, Electrical Engineering Department**