Adapting Transformer-XL to QANet for SQuAD 2.0
Lorraine Zhang [lz2017]@stanford.edu

**Motivation**
- Explore a novel approach to reading comprehension system
- Experiment with deep learning techniques for question answering
- Improve QANet performance on SQuAD 2.0

**DATA**
- Source: https://github.com/chrischute/squad.git
- Datasets = {Training set: 129,941 examples, Dev set: 6078 examples, Test set: 5915 examples}
- Pretrained GloVectors: 300-dimensional embeddings trained on CommonCrawl 840B corpus.

**RESULTS**

<table>
<thead>
<tr>
<th>Models</th>
<th>F1</th>
<th>EM</th>
<th>Epochs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline BiDAF</td>
<td>61</td>
<td>57.45</td>
<td>30</td>
</tr>
<tr>
<td>Baseline QANet, SQ1.0</td>
<td>76.2</td>
<td>66.3</td>
<td>30</td>
</tr>
<tr>
<td>QANet, dev set</td>
<td>68.46</td>
<td>64.81</td>
<td>30</td>
</tr>
<tr>
<td>QANet, test set</td>
<td>65.18</td>
<td>61.56</td>
<td>30</td>
</tr>
<tr>
<td>QANet+Transformer-XL</td>
<td>64.87</td>
<td>61.75</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 1: F1, EM scores, non-PCE

**PROBLEM DEFINITION**

- **Challenge:**
  - Answer questions correctly in longer context on a reading comprehension system
  - Many models such as QANet are limited by fixed_length dependency

- **Evaluation Metric:**
  - EM score:
    - Exact Match to ground truth answer
    - Harmonic mean of precision and recall
  - F1 score:
    - F1 = 2 x prediction x recall / (precision + recall)

**APPROACH**

**QANet-XL Model:**
- Cache memory and feedback to EncoderBlock
- Only use convolution and self attention
- Adam optimizer with warm-up rate

**Transformer-XL Techniques used:**
- Recurrence Mechanism
  - Cache and reuse hidden states as memory for the current state
- Relative Positional Encoding Scheme
  - Only encode the relative positional information in the hidden states
- New Variables Introduced:
  1. mems: previous state
  2. r: relative positional encoding
  3. r_r_bias
  4. r_w_bias

- Difference in Self Attention in EncoderBlock:
  - $A_t^{q_i} = q_i^T k_j = \sum_{r \in R} w_{a,r} E_{s_{r \in R}} + \sum_{r \in R} w_{a,R} R_{s_{r \in R}} + u^T W_q W_k U_j + v^T W_k W_R U_j$, (Transformer)
  - $A_t^{q_i} = q_i^T k_j = \sum_{r \in R} w_{a,R} E_{s_{r \in R}} + \sum_{r \in R} w_{a,R} R_{s_{r \in R}} + u^T W_q W_k U_j + v^T W_k W_R U_j$, (Transformer XL)

**CONCLUSIONS**

- Both QANet and QANet-XL outperformed the baseline BiDAF in F1 and EM scores.
- Recurrence mechanism increased memory requirement on hardware noticeably.
- QANet-XL underperformed vanilla QANet.
- Limitations on time and hardware prevented adequate training of QANet-XL.
- Had to use different hyperparameters due to lack of available memory.
- QANet-XL: 46 hidden size, 4 heads vs QANet: 128 hidden size, 8 on hidden size
- Lower NLL and steeper trajectory of F1/EM of QANet-XL indicate its promise.
- Datasize and character embedding dimension impacted F1 and EM scores.
- Underperformed original QANet paper which used 3x augmented dataset and 200-dimension character embedding vs 96 characters in this project.

**FUTURE WORK**

- Increase dataset size to get better pre-train model.
- Increase hidden size and number of heads on higher performance hardware with more memory.

**REFERENCES**