Problem and Motivation

Goal
• Question and answering is a machine comprehension task in which a passage and a question are provided to a machine, and the machine must provide the answer.
• The recent release of SQuAD 2.0 has expanded the task by including unanswerable questions.

Motivation
• We attempt to utilize the recent Transformer-XL architecture, which leverage segment-level recurrence mechanism to perform better on both long and short sequences [2].
• We use QANet as our starting point [1], but we also look at other recent models for inspiration.

Task
In this project, we build a deep learning model to perform machine comprehension on the SQuAD 2.0 dataset. We utilize techniques such as multi-level attention, self-attention, fuse representation, and Transformer-XL hidden states [5][4].

Embedding Layer
We used the following embeddings as the input of our model:
• GLoVE
• Character-level embedding with convolutions, max-pooling, and highway layer activation.
• Parts of Speech (PoS)
• NeR (Nearest Entity recognition)
• Exactly matching tokens
• Lower case match
• Lemma matches
• Term frequency – inverse document frequency (tf-idf)

The additional embeddings are extracted with the aid of spaCy.

Additionally, we utilize the final hidden states of the Transformer-XL in our model. Where Transformer-XL is pretrained on the WikiText Long Term Dependency Language Modeling Dataset.

Code Source for transformer-XL: https://github.com/huggingface/pytorch-pretrained-BERT

Architectures Implementation

Results

<table>
<thead>
<tr>
<th>Model (dev)</th>
<th>F1</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>55.6</td>
<td>58.9</td>
</tr>
<tr>
<td>QANet</td>
<td>64.2</td>
<td>67.9</td>
</tr>
<tr>
<td>Final Model</td>
<td>66.3</td>
<td>70.2</td>
</tr>
</tbody>
</table>

• We used the AdaMax optimizer with a decaying learning rate starting at 0.002.
• We used a dropout rate of 30% across all layers, including the embedding layer
• Whereas QANet completed a single epoch within 20 minutes, the final model took over 3 hours per epoch (trained on a RTX 2080ti).
• This massive reduction in speed is likely caused by the removal of convolutional parallelism and the inclusion of Transformer-XL
• We achieved a final test set result of EM and F1 of 64.3 and 68.1

References