**Problem**

- Machine reading comprehension and question answering is an essential task in natural language processing. It is always challenging since it requires a comprehensive understanding of natural languages and the ability to do further inference and reasoning.
- Recently, Pre-trained Contextual Embeddings (PCE) models like ELMo and BERT have attracted lots of attention due to their great performance in a wide range of NLP tasks.
- In this project, we picked up BERT model and tried to fine-tune it with additional task-specific layers to improve its performance on Stanford Question Answering Dataset (SQuAD 2.0).

**Data**

- We used Stanford Question Answering Dataset (SQuAD 2.0) to train and evaluate our models.
- Samples in this dataset include (question, answer, context paragraph) tuples.
- The paragraphs are from Wikipedia. The questions and answers were crowdsourced using Amazon Mechanical Turk.
- We have around 150k questions in total, and roughly half of the questions are not answerable.
- If a question is answerable, the answer is guaranteed to be a continuous span in the context paragraph.

**Approach**

- **Our main idea** is to add an encoder-decoder architecture on top of the BERT model. This idea comes from the computer vision area. For multi-view synthesis task, we can use a general auto-encoder to generate the sketch of other views and an additional auto-encoder for texture level reconstruction.

- **Modules on Top of BERT**
  - **BERT** is a multi-layer bidirectional encoder. By training deep transformers on carefully designed bidirectional language modeling task, we can get the pre-trained BERT representations that we are going to fine-tune later with additional output layers.
  - **Encoder/Decoder**: RNN Encoder (Bi-LSTM, GRU): try to integrate temporal dependencies between time-steps of the output tokenized sequence better. CNN Encoder: 2D-Convolution on the dimension of seq_len and hidden_state. Extract the relationship of nearby word embeddings in the sequence.
  - **Self-Attention**: Optimization of networks with virtually arbitrary depth. By applying a gating mechanism, a neural network can have paths along which information can flow across several layers without attenuation. It serves as multi-layer state transitions in RNN to allow the network to adaptively copy or transform representations.

**Results and Analysis**

- **Results Summary**
  - Architecture on Top of BERT | F1 | EM
  - BERT-base pytorch Implementation | 76.75 | 74.85
  - BERT-base Tensorflow Implementation | 76.07 | 72.80
  - GRU Encoder + Self-attention + GRU Decoder + BERT-SQUAD-Out | 75.59 | 65.87
  - BiLSTM Encoder + BiLSTM-Out | 76.37 | 73.05
  - CNN Encoder + Self-attention +BERT-SQUAD-Out | 75.49 | 73.23
  - CNN Encoder + BERT-SQUAD-Out | 76.56 | 73.64
  - GRU Encoder + GRU Decoder + BERT-SQUAD-Out | 75.85 | 73.71
  - BiLSTM Encoder + BiLSTM Decoder + Highway + BERT-SQUAD-Out | 77.07 | 73.87
  - BiLSTM Encoder + Highway + BERT-SQUAD-Out | 77.41 | 74.32
  - BiLSTM Encoder + Highway + BiLSTM Decoder + BERT-SQUAD-Out | 77.06 | 74.87
  - BiLSTM Encoder + BiLSTM Decoder + Highway + BERT-SQUAD-Out | 77.96 | 74.98
  - Ensemble of 11 and 7 | 78.35 | 75.60
  - Ensemble of 11 and 7 and BERT large case model | 79.44 | 76.96

**Error Analysis Example**

<table>
<thead>
<tr>
<th>Questions</th>
<th>model 1</th>
<th>model 3</th>
<th>model 11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who made fun of the Latin language?</td>
<td>No Answer</td>
<td>Geoffroy Chausser</td>
<td>No Answer</td>
</tr>
<tr>
<td>Who led Issacs troops to Cyprus?</td>
<td>No Answer</td>
<td>Guy de Lusignan</td>
<td>No Answer</td>
</tr>
<tr>
<td>Who began a program of church reform in the 1100s?</td>
<td>No Answer</td>
<td>the dukes</td>
<td>No Answer</td>
</tr>
</tbody>
</table>

**Conclusion**

- We added several components on top of the BERT model as task-specific layers and analyzed their performance compared to BERT baseline model in great details.
- Our best model so far implements BiLSTM Encoder + BiLSTM Decoder + Highway + BERT-SQUAD-Out as the output architecture on BERT uncased base model, and it achieves an F1 score of 77.96 on the Dev set.
- With ensemble technique, we finally achieved an F1 score of 79.44 on the Dev Set and 77.827 on the Test Set, ranked 12th on the 224N leaderboard.

**Reference**
