The Death of Feature Engineering? — BERT with Linguistic Features on SQuAD 2.0

Yue Zhang (yzhang16), Jiawei Li (jiaweili)

Background

- Machine reading comprehension is an essential NLP task, it is useful both for application and as a measure of how good NLP models understand given text.
- Input: A pair of context and query
- Prediction: The corresponding answer to query.

Dataset

- Dataset: Stanford Question Answering Dataset 2.0 [4], extending SQuAD 1.0 by adding questions that have no answer in the given text.
- train set (129,941 examples): All taken from the official SQuAD 2.0 training set.
- dev set (6078 examples): Roughly half of the official dev set, randomly selected.
- test set (5921 examples): The remaining examples from the official dev set, plus hand-labeled examples.

Motivation

- State-of-the-art model (BERT) performs well, close to human level, but still have some NLU errors.
- We propose incorporating linguistic features to help.

Framework

- Input: a pair of sentences: context c and question q
- Output: answer for the question (a span on the context sentences): start and end token index: x_start, x_end
- Metrics: Exact Match (EM) score and F1 score
- Model: BERT [3, 4] and Linguistic Feature model

BERT Model

- Input features from tokenizer:
  
  (input_idx, mask, segment) = tokenizer(c, q)

- Sequential Embedding Features from BERT:
  
  seq_out = BERT(input_idx, mask, segment)

Linguistic Feature Model

- Linguistic features from Linguistic model:
  
  ling_out = linguistic_model(c, q)

Output Layer

- Concatenate the output from both model:
  
  output_logits = output_layer(seq_out, ling_out)

Linguistic Features

- 4 linguistic features are extracted for each token with NLP package SpaCy [2], the first 3 features encoded as integers.
- NER: Name entity label
- POS: The part-of-speech tag
- DEP: Syntactic dependency, token relationship
- STOP: Is the token part of a stop list, 0/1 vector

Table 1: Experiments Results for Single Model

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Dev Set</th>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT (large)</td>
<td>78.51/81.20</td>
<td>76.55/79.97</td>
</tr>
<tr>
<td>BERT (base)</td>
<td>71.59/74.72</td>
<td>73.76/76.86</td>
</tr>
<tr>
<td>BiDAF</td>
<td>49.07/50.29</td>
<td>50.75/51.94</td>
</tr>
<tr>
<td>BiLSTM</td>
<td>47.54/49.78</td>
<td>48.52/50.76</td>
</tr>
<tr>
<td>LSTM</td>
<td>44.00/46.25</td>
<td>45.00/47.25</td>
</tr>
</tbody>
</table>

Table 2: Model Predictions

<table>
<thead>
<tr>
<th>Question</th>
<th>Reference</th>
<th>BERT-feature</th>
<th>Dev Set</th>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>What non-Chinese empire did the Yuan dynasty succeed?</td>
<td>Mongol</td>
<td>No Answer</td>
<td>76.55/79.97</td>
<td></td>
</tr>
<tr>
<td>Which tribes did Genghis Khan fight against?</td>
<td>Mongol and Turkic</td>
<td>No Answer</td>
<td>76.55/79.97</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Confusion Matrix for the Existence of the Answer

<table>
<thead>
<tr>
<th></th>
<th>Answer</th>
<th>No Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer</td>
<td>1456</td>
<td>454</td>
</tr>
<tr>
<td>No Answer</td>
<td>1456</td>
<td>1612</td>
</tr>
</tbody>
</table>

Conclusion and Future Work

- Adding features help improve performance of BERT base, no significant improvement for BERT large.
- We conclude "Feature engineering is not dying", especially when computational resources are not very cheap today.
- In future, we are interested in modifying the architecture and the loss function to get better results on the Answer / No Answer classification problem, multitasking learning is also a good candidate for making improvement on that.

References