Transformer Based Question Answering Model

Emma Chen, Jennifer She

Problem

- Study the performance of attention-based models (inspired by Transformer and QANet) in solving the SQuAD 2.0 Question Answering Challenge.
- Reproduce QANet as a competitive alternative to the LSTM-based baseline model BiDAF.
- Experiment with variants of QANet (and BiDAF).

Approach

1. BiDAF + Char Emb: Augment BiDAF baseline with character embeddings
2. QANet + Char Emb: Reproduce QANet
3. QANet + Char Emb + Trainable Word Emb: Allow word vectors to be trainable
4. QANet + Char Emb + Conv Layers: Replace depthwise separable conv layers with regular convolutions
5. QANet + Char Emb + Recurrence: Divide context into segments and feed sequentially into model encoding

Data/Task

Data: SQuAD 2.0 dataset modified and provided by the course with train, dev and test set
- 140k examples of context-question-answer triples

Task: given a pair of context and question, correctly predict the answer
- ~5 context-question pairs do not have an answer: the model should correctly predict N/A
- ~5 context-question pairs have answers: the model should correctly predict the starting + ending position of the answer in the context

Example:

Context: In 1066, Duke William II of Normandy conquered England killing King Harold II at the Battle of Hastings. The invading Normans and their descendants replaced the Anglo-Saxons as the ruling class of England. The nobility of England were part of a single Norman culture and many had lands on both sides of the channel. Early Norman kings of England, at Duke of Normandy, owed homage to the King of France for their land on the continent. This considered England to be a most important holding (it brought with it the title of King—an important status symbol).

Question: When did King Harold II conquer England?

Prediction: 1066

Answer: N/A

QANet + Char Emb tends to predict answers to questions that have no answer—especially "when" and "what" questions that appear easy.

Analysis

1. BiDAF + Char Emb and QANet + Char Emb are comparable
2. BiDAF + Char Emb + Conv Layers works better than training data (lower dev NLL, higher dev F1)
3. BiDAF + Char Emb + Recurrence leads slightly better performance (lower dev NLL and higher dev F1) than dividing context into 2 segments

Question: Why did OPEC block oil deliveries to the United States?

Context: In response to American aid to Israel, on October 16, 1973, OPEC raised the posted price of oil by 70%, to $5.11 a barrel. The following day, oil ministers agreed to block all oil deliveries to the US as a "principal hostile country".

Answer: American aid to Israel

Prediction: principal hostile country

QANet + Char Emb tends to favor near-true answers in question answering.

Results

<table>
<thead>
<tr>
<th>Model</th>
<th>F1</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiDAF + Char Emb</td>
<td>64.9</td>
<td>61.4</td>
</tr>
<tr>
<td>QANet + Char Emb</td>
<td>65.1</td>
<td>61.2</td>
</tr>
<tr>
<td>QANet + Char Emb + Trainable Word Emb</td>
<td>54.8</td>
<td>52.3</td>
</tr>
<tr>
<td>QANet + Char Emb + Conv Layers</td>
<td>51.8</td>
<td>51.6</td>
</tr>
<tr>
<td>QANet + Char Emb + Recurrence</td>
<td>61.5</td>
<td>57.8</td>
</tr>
</tbody>
</table>

Conclusions

1. Adding character embeddings improves performance of both BiDAF and QANet
2. QANet + Char Emb is competitive with BiDAF + Char Emb
3. Making word embeddings trainable leads to overfitting
4. Replacing depthwise separable convolutions with regular convolutions leads to worse learning
5. Incorporating recurrence into QANet model encoding works reasonably well (also better with smaller segments / more recurrence)

References