A Co-Matching Model for Multi-choice Reading Comprehension

Shuohang Wang¹, Mo Yu², Shiyu Chang², Jing Jiang¹

¹Singapore Management University
²IBM Research
Reading Comprehension

- The task: to answer questions given a passage of text
- Datasets
  - CNN/Daily Mail [Hermann et al. 2015]
  - Children’s Book Test [Hill et al. 2016]
  - SQuAD [Rajpurkar 2016]
  - MCTest [Richardson et al. 2013]
  - RACE [Lai et al. 2017]
  - NarrativeQA [Kocisky et al. 2018]
RACE

• Passage: My father wasn’t a king, he was a taxi driver, but … … I met Blandy at a party and he asked if I’d like to buy the island. Of course I said yes but I had no money—I was just an art teacher. I tried to find some business partners, who all thought I was crazy. So I sold some of my possessions, put my savings together and bought it … …

• Question: How did the author get the island?
  • a. It was a present from Blandy.
  • b. The king sold it to him.
  • c. He bought it from Blandy.
  • d. He inherited from his father.

Challenge: to jointly model passage, question and candidate answers
Related Work

• Converted to sequence pair matching [Yin et al., 2016]
  • Each candidate answer is concatenated with the question
  • The concatenated sequences are matched against the passage

Limitation:
Question and answers are not clearly separated. Interaction information between a question and an answer is lost.
Related Work

• Matching sequences pair by pair [Lai et al., 2017]
  • Match passage and question first
  • Then this representation is used to match candidate answers

Limitation:
Matching P & Q may not give meaningful representations for questions like “Which statement of the following is true?”
Our Solution

• Co-match each sentence in the passage with the question and the candidates answers separately.

• Make use of the alignments between sequences as follows:

  Question: How did the author get the island?

  Passage

  Candidate Answer: He bought it from Blandy

• Hierarchically aggregate the co-matching representations of (sentence, question, answer) triplets for final scoring.
Co-Matching

• For every word in sentence, we match it with the attention-weighted vectors computed based on the question and the candidate answer, respectively.

Question: How did the author *get* the *island*?

Passage

Candidate Answer: He *bought* it from *Blandy*
Framework

Co-Matching states

Matching states

Attention-weighted vectors

Attention weights

LSTM states

Embedding

Question

1st Sentence of Passage

Candidate answer
Framework

Hierarchical LSTM

Co-Matching states

Matching states

Attention-weighted vectors

Attention weights

LSTM states

Embedding

Question 1st Sentence of Passage Candidate answer 2nd Sentence of Passage Nth Sentence of Passage

Representation for ranking candidates
Experiments

Our **Hier-Co-Matching** achieved the best performance compared with previous work.

We studied two key factors:
(1) the _co-matching_ module
(2) the _hierarchical_ aggregation approach

<table>
<thead>
<tr>
<th></th>
<th>RACE-M</th>
<th>RACE-H</th>
<th>RACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>24.6</td>
<td>25.0</td>
<td>24.9</td>
</tr>
<tr>
<td>Sliding Window</td>
<td>37.3</td>
<td>30.4</td>
<td>32.2</td>
</tr>
<tr>
<td>Stanford AR</td>
<td>44.2</td>
<td>43.0</td>
<td>43.3</td>
</tr>
<tr>
<td>GA</td>
<td>43.7</td>
<td>44.2</td>
<td>44.1</td>
</tr>
<tr>
<td>ElimiNet</td>
<td>-</td>
<td>-</td>
<td>44.7</td>
</tr>
<tr>
<td>HAF</td>
<td>45.3</td>
<td>47.9</td>
<td>47.2</td>
</tr>
<tr>
<td>MUSIC</td>
<td>51.5</td>
<td>45.7</td>
<td>47.4</td>
</tr>
<tr>
<td>Hier-Co-Matching</td>
<td>55.8*</td>
<td>48.2*</td>
<td>50.4*</td>
</tr>
</tbody>
</table>
  - Hier-Aggregation | 54.2   | 46.2   | 48.5  |
  - Co-Matching       | 50.7   | 45.6   | 46.4  |
| Turkers             | 85.1   | 69.4   | 73.3  |
| Ceiling             | 95.4   | 94.2   | 94.5  |
Conclusions

• We proposed a hierarchical co-matching model for answering multi-choice reading comprehension questions.

• We showed that our model could achieve state-of-the-art performance on the RACE dataset.

• There is still much room for improvement on RACE given the low absolute performance.
  • Latest results by OpenAI: 59%

Paper:  
Code:
References


