Attention-over-Attention Neural Networks for Reading Comprehension

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu and Guoping Hu

Joint Laboratory of HIT and IFLYTEK Research (HFL), Beijing, China
2017-08-01
ACL2017, Vancouver, Canada
OUTLINE

• Introduction: Cloze-style Reading Comprehension

• Related Works

• Attention-over-Attention Reader (AoA Reader)

• N-best Re-ranking Strategy

• Experiments & Analysis

• Conclusions & Future Works
INTRODUCTION

- *Machine Reading Comprehension (MRC)* is to read and comprehend a given article and answer the questions based on it, which has become enormously popular in recent few years.

- The related datasets and algorithms are mutually benefitted.
  - From cloze-style to sentence-style
  - From simple model to complex model

- In this paper, we focus on solving the cloze-style RC problem.
INTRODUCTION

• Key components in RC

→ Document

• Query

• Candidates

• Answer

*Example is chosen from the MCTest dataset (Richardson et al., 2013)
INTRODUCTION

• Key components in RC

• Document

→ Query

• Candidates

• Answer

James the Turtle was always getting in trouble. Sometimes he'd reach into the freezer and empty out all the food. Other times he'd sled on the deck and get a splinter. His aunt Jane tried as hard as she could to keep him out of trouble, but he was sneaky and got into lots of trouble behind her back.

One day, James thought he would go into town and see what kind of trouble he could get into. He went to the grocery store and pulled all the pudding off the shelves and ate two jars. Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn't pay, and instead headed home.

His aunt was waiting for him in his room. She told James that she loved him, but he would have to start acting like a well-behaved turtle.

After about a month, and after getting into lots of trouble, James finally made up his mind to be a better turtle.

1) What is the name of the trouble making turtle?
A) Fries
B) Pudding
C) James
D) Jane

*Example is chosen from the MCTest dataset (Richardson et al., 2013)
INTRODUCTION

• Key components in RC
  - Document
  - Query

→ Candidates

• Answer

*Example is chosen from the MCTest dataset (Richardson et al., 2013)
INTRODUCTION

• Key components in RC
  • Document
  • Query
  • Candidates

→ Answer

James the Turtle was always getting in trouble. Sometimes he’d reach into the freezer and empty out all the food. Other times he’d sled on the deck and get a splinter. His aunt Jane tried as hard as she could to keep him out of trouble, but he was sneaky and got into lots of trouble behind her back.

One day, James thought he would go into town and see what kind of trouble he could get into. He went to the grocery store and pulled all the pudding off the shelves and ate two jars. Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn’t pay, and instead headed home.

His aunt was waiting for him in his room. She told James that she loved him, but he would have to start acting like a well-behaved turtle.

After about a month, and after getting into lots of trouble, James finally made up his mind to be a better turtle.

1) What is the name of the trouble making turtle?
A) Fries  
B) Pudding  
C) James  
D) Jane

*Example is chosen from the MCTest dataset (Richardson et al., 2013)
INTRODUCTION

• Specifically, in cloze-style RC

  • Document: the same as the general RC

  • Query: a sentence with a blank

  • Candidate (optional): several candidates to fill in

  • Answer: a single word that exactly match the query (the answer word should appear in the document)

*Example is chosen from the CNN dataset (Hermann et al., 2015)
INTRODUCTION

• CBT dataset (Hill et al., 2015)
**RELATED WORKS**

- Predictions on full vocabulary
  - Attentive Reader (Hermann et al., 2015)
  - Stanford AR (Chen et al., 2016)
- Pointer-wise predictions (Vinyals et al., 2015)
  - Attention Sum Reader (Kadlec et al., 2016)
  - Consensus Attention Reader (Cui et al., 2016)
  - Gated-attention Reader (Dhingra et al., 2017)
ATTENTIVE READER

• Teaching Machines to Read and Comprehend (Hermann et al., 2015)
A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task (Chen et al., 2016)
ATTENTION SUM READER

- Text Understanding with the Attention Sum Reader Network (Kadlec et al., 2016)
Consensus Attention Reader

- Consensus Attention-based Neural Networks for Chinese Reading Comprehension (Cui et al., 2016)
GATED-ATTENTION READER

- Gated-Attention Reader for Text Comprehension (Dhingra et al., 2016)
AoA Reader

• Primarily motivated by AS Reader (Kadlec et al., 2016) and CAS Reader (Cui et al., 2016)
  • Introduce matching matrix for indicating doc-query relationships
  • Mutual attention: doc-to-query and query-to-doc
  • Instead of using heuristics to combine individual attentions, we place another attention to dynamically assign weights to the individual ones
• Some of the ideas in our work has already been adopted in the follow-up works not only in cloze-style RC but also other types of RC (such as SQuAD).
AoA Reader

• Model architecture at a glance

\[
P(\text{"Mary"}|D,Q) = \sum_{i=1}^{\text{let}(\text{"Mary"}|D)} s_i = s_j + s_k
\]
AoA Reader

• Contextual Embedding

• Transform document and query into contextual representations using word-embeddings and bi-GRU units

\[
e(x) = W_e \cdot x, \text{ where } x \in D, Q \quad (1) \\
\hat{h}_s(x) = \text{GRU}(e(x)) \quad (2) \\
\hat{h}_s(x) = \text{GRU}(e(x)) \quad (3) \\
h_s(x) = [\hat{h}_s(x); \hat{h}_s(x)] \quad (4)
\]
**AoA Reader**

- **Pair-wise Matching Score**
  - Calculate similarity between each document word and query word
  - For simplicity, we just calculate dot product between document and query word

\[ M(i, j) = h_{doc}(i)^T \cdot h_{query}(j) \] (5)
AoA Reader

- Individual Attentions
  - Calculate document-level attention with respect to each query word

\[ \alpha(t) = \text{softmax}(M(1, t), ..., M(|D|, t)) \]  \hspace{1cm} (6)

\[ \alpha = [\alpha(1), \alpha(2), ..., \alpha(|Q|)] \]  \hspace{1cm} (7)
**AoA Reader**

- **Attention-over-Attention**

- Dynamically assign weights to individual doc-level attentions

\[
\beta(t) = \text{softmax}(M(t, 1), \ldots, M(t, |Q|)) \quad (8)
\]

\[
\beta = \frac{1}{n} \sum_{t=1}^{|D|} \beta(t) \quad (9)
\]

\[
s = \alpha^T \beta \quad (10)
\]
**Final Predictions**

- Pointer Network (Vinyals et al., 2015)
- Apply sum-attention mechanism (Kadlec et al., 2016) to get the final probability of the answer

\[
P(w|D, Q) = \sum_{i \in I(w, D)} s_i, \ w \in V \tag{11}
\]

\[
\mathcal{L} = \sum_{i} \log(p(x)) , x \in \mathcal{A} \tag{12}
\]
**AoA Reader**

- **An intuitive example:** Let say this is a story about `Tom bought a diamond ring for his beloved girl friend…` 

<table>
<thead>
<tr>
<th></th>
<th>Tom</th>
<th>loves</th>
<th>&lt;blank&gt;</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query-level Attention</strong></td>
<td>0.5</td>
<td>0.3</td>
<td>0.15</td>
<td>0.05</td>
</tr>
</tbody>
</table>
| **Candidate Answers**    | Mary = 0.6  
  diamond = 0.3  
  beside = 0.1 | Mary = 0.3  
  diamond = 0.5  
  beside = 0.2 | Mary = 0.4  
  diamond = 0.4  
  beside = 0.2 | Mary = 0.2  
  diamond = 0.4  
  beside = 0.4 |
| **Average Score**
  (Cui et al., 2016)       | Mary  
  = (0.6+0.3+0.4+0.2) / 4 = 0.375  
  diamond  
  = (0.3+0.5+0.4+0.4) / 4 = 0.400  
  beside  
  = (0.1+0.2+0.2+0.4) / 4 = 0.225 |
| **Weighted Score**
  (This work)              | Mary  
  = 0.6*0.5+0.3*0.3+0.4*0.15+0.2*0.05 = 0.460  
  diamond  
  = 0.3*0.5+0.5*0.3+0.4*0.15+0.4*0.05 = 0.380  
  beside  
  = 0.1*0.5+0.2*0.3+0.2*0.15+0.4*0.05 = 0.160 |
RE-RANKING

• N-best re-ranking strategy for cloze-style RC

• Mimic the process of double-checking, in terms of fluency, grammatical correctness etc.

• Main idea: Re-fill the candidate answer into the blank of query to form a complete sentence and using additional features to score the sentences
**RE-RANKING**

- **Procedure of re-ranking**
  - Generate candidate answers: N-best decoding
  - Refill the candidate into query
  - Scoring with additional features: mainly LM features
  - Feature weight tuning: using K-Best MIRA algorithm (Cherry and Foster, 2012)
  - Re-scoring and Re-ranking
Re-ranking

• Features that used in re-ranking

  • Global LM: trained on document part of training data
    - Word LM: 8-gram LM using SRILM tool (Stolcke, 2002)
    - Word-class LM: 1,000 word classes using mkcls tool (Josef Och, 1999)
  • Local LM: trained on document part of test-time data sample-by-sample
EXPERIMENTS

• Dataset
  
  • CNN(Hermann et al., 2015), CBT-NE/CN (Hill et al., 2015)

• Hyper-parameters
  
  • Embedding: uniform distribution with l2-regularization, dropout 0.1
  
  • Hidden Layer: bi-GRU
  
  • Optimization: Adam(lr=0.001), gradient clipping 5, batch 32

• Implementation: Keras (Chollet, 2015) + Theano (Theano Development Team, 2016)
EXPERIMENTAL RESULTS

- Single model performance
  - Significantly outperform previous works
  - Re-ranking strategy could substantially improve performance

<table>
<thead>
<tr>
<th>Model</th>
<th>CNN Valid</th>
<th>News Valid</th>
<th>CBTTest NE Valid</th>
<th>CBTTest CN Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep LSTM Reader (Hermann et al., 2015)</td>
<td>55.0</td>
<td>57.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Attentive Reader (Hermann et al., 2015)</td>
<td>61.6</td>
<td>63.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Human (context+query) (Hill et al., 2015)</td>
<td>-</td>
<td>-</td>
<td>81.6</td>
<td>81.6</td>
</tr>
<tr>
<td>MemNN (window + self-sup.) (Hill et al., 2015)</td>
<td>63.4</td>
<td>66.8</td>
<td>70.4</td>
<td>66.6</td>
</tr>
<tr>
<td>AS Reader (Kadlec et al., 2016)</td>
<td>68.6</td>
<td>69.5</td>
<td>73.8</td>
<td>68.6</td>
</tr>
<tr>
<td>CAS Reader (Cui et al., 2016)</td>
<td>68.2</td>
<td>70.0</td>
<td>74.2</td>
<td>69.2</td>
</tr>
<tr>
<td>Stanford AR (Chen et al., 2016)</td>
<td>72.4</td>
<td>72.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GA Reader (Dhingra et al., 2016)</td>
<td>73.0</td>
<td>73.8</td>
<td>74.9</td>
<td>69.0</td>
</tr>
<tr>
<td>Iterative Attention (Sordoni et al., 2016)</td>
<td>72.6</td>
<td>73.3</td>
<td>75.2</td>
<td>68.6</td>
</tr>
<tr>
<td>EpiReader (Trischler et al., 2016)</td>
<td>73.4</td>
<td>74.0</td>
<td>75.3</td>
<td>69.7</td>
</tr>
<tr>
<td>AoA Reader</td>
<td>73.1</td>
<td>74.4</td>
<td>77.8</td>
<td>72.0</td>
</tr>
<tr>
<td>AoA Reader + Reranking</td>
<td>-</td>
<td>-</td>
<td>79.6</td>
<td>74.0</td>
</tr>
<tr>
<td>MemNN (Ensemble)</td>
<td>66.2</td>
<td>69.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>AS Reader (Ensemble)</td>
<td>73.9</td>
<td>75.4</td>
<td>74.5</td>
<td>70.6</td>
</tr>
<tr>
<td>GA Reader (Ensemble)</td>
<td>76.4</td>
<td>77.4</td>
<td>75.5</td>
<td>71.9</td>
</tr>
<tr>
<td>EpiReader (Ensemble)</td>
<td>-</td>
<td>-</td>
<td>76.6</td>
<td>71.8</td>
</tr>
<tr>
<td>Iterative Attention (Ensemble)</td>
<td>74.5</td>
<td>75.7</td>
<td>76.9</td>
<td>72.0</td>
</tr>
<tr>
<td>AoA Reader (Ensemble)</td>
<td>-</td>
<td>-</td>
<td>78.9</td>
<td>74.5</td>
</tr>
<tr>
<td>AoA Reader (Ensemble + Reranking)</td>
<td>-</td>
<td>-</td>
<td>80.3</td>
<td>75.6</td>
</tr>
</tbody>
</table>
Experimental Results

• Single model performance

• Introducing attention-over-attention mechanism instead of using heuristic merging function (Cui et al., 2016) may bring significant improvements
EXPERIMENTAL RESULTS

• Ensemble performance

• We use greedy ensemble approach of 4 models trained on different random seed

• Significant improvements over state-of-the-art systems
RE-RANKING ABLATIONS

- Calculate weight proportion between global and local LMs
  \[ \eta = \frac{LM_{global} + LM_{wc}}{LM_{local}} \]

- Observations
  - NE category seems to be more dependent on local LM
  - CN category seems to be more dependent on global LM

<table>
<thead>
<tr>
<th></th>
<th>CBTest NE Valid</th>
<th>CBTest NE Test</th>
<th>CBTest CN Valid</th>
<th>CBTest CN Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>AoA Reader</td>
<td>77.8</td>
<td>72.0</td>
<td>72.2</td>
<td>69.4</td>
</tr>
<tr>
<td>+Global LM</td>
<td>78.3</td>
<td>72.6</td>
<td>73.9</td>
<td>71.2</td>
</tr>
<tr>
<td>+Local LM</td>
<td>79.4</td>
<td>73.8</td>
<td>74.7</td>
<td>71.7</td>
</tr>
<tr>
<td>+Word-class LM</td>
<td>79.6</td>
<td>74.0</td>
<td>75.7</td>
<td>73.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>CBTest NE</th>
<th>CBTest CN</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>0.64</td>
<td>0.20</td>
</tr>
<tr>
<td>Global LM</td>
<td>0.16</td>
<td>0.10</td>
</tr>
<tr>
<td>Word-class LM</td>
<td>0.04</td>
<td>0.39</td>
</tr>
<tr>
<td>Local LM</td>
<td>0.16</td>
<td>0.31</td>
</tr>
<tr>
<td>RATIO (\eta)</td>
<td>1.25</td>
<td>1.58</td>
</tr>
</tbody>
</table>
QUANTITATIVE ANALYSIS

• Accuracy vs. Length of Document

  • AoA Reader shows consistent improvements over AS Reader on different length of document

  • Improvements become larger when the length of document increases, indicating that our model could better handle the long documents
QUANTITATIVE ANALYSIS

• Accuracy vs. Frequency of answer
  • Most of the answers are the top frequent word among candidates
  • Tend to choose either high or low frequency word

Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, G. Hu
CONCLUSIONS & FUTURE WORK

• Conclusion

  • Propose a novel mechanism called “Attention-over-Attention” to dynamically assign weights to the individual attentions
  
  • Two-way attention: adopt both doc-to-query and query-to-doc attentions for final predictions
  
  • Experimental results show significant improvements over various state-of-the-art systems

• Future Work

  • Investigate more complex attention mechanism via adopting external knowledge
  
  • Look into the questions that need comprehensive reasoning over several sentences
**EXTENSION: INTERACTIVE AOA READER**

- **Interactive AoA Reader**

  - As a step further of our work, we’ve refined our model as ‘interactive’, which dynamically and progressively filter the context for question answering.

  - Shows state-of-the-art performance and ranked **No.1** in Stanford SQuAD Task (Rajpurkar et al., 2016)

---

**Leaderboard**

Since the release of our dataset, the community has made rapid progress! Here are the ExactMatch (EM) and F1 scores of the best models evaluated on the test and development sets of v1.1. Will your model outperform humans on the QA task?

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interactive AoA Reader (ensemble)</td>
<td>77.845</td>
<td>85.297</td>
</tr>
</tbody>
</table>

REFERENCES


• François Chollet. 2015. Keras. https://github.com/fchollet/keras.


REFERENCES


Thank you!
Enjoy your time in Vancouver!

Contact: me [at] ymcui [dot] com